

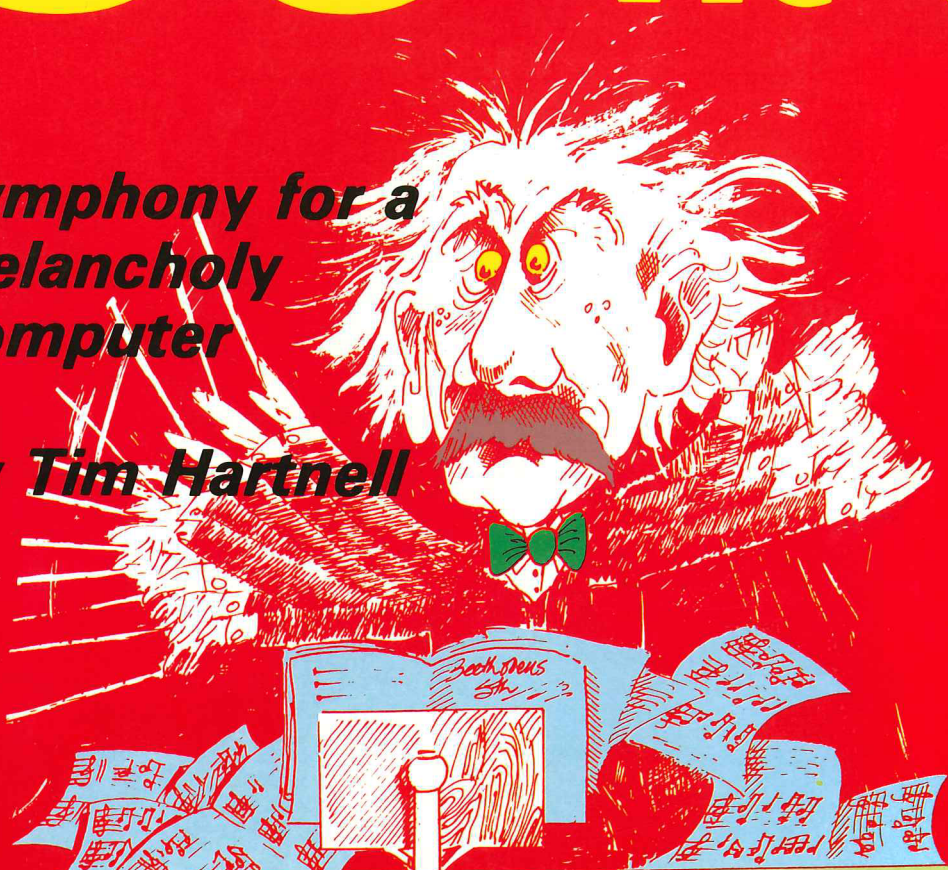
# 50

**OUTSTANDING  
PROGRAMS FOR THE**

# **VIC-20**

## ***Symphony for a Melancholy Computer***

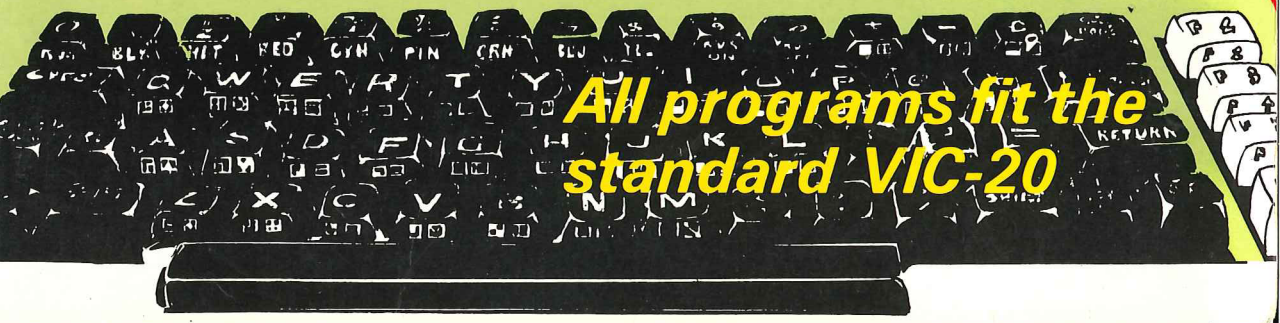
***By Tim Hartnell***



Commodore VIC-20

***Second edition***

***All programs fit the  
standard VIC-20***







# **SYMPHONY FOR A MELANCHOLY COMPUTER**

**Tim Hartnell**

**INTERFACE**  
PUBLICATIONS

46 Earls Court Road, London W8 6EJ



*For Keith and Alan*



Symphony for a Melancholy Computer

*First published in Great Britain by*

INTERFACE PUBLICATIONS,  
44-46 Earls Court Road,  
LONDON, W8 6EJ

(c) Hartnell, 1982

First edition – June, 1982

Second edition – November, 1982

ISBN 0 907563 14 7

All rights reserved. No part of this publication may be stored in a retrieval system, reproduced, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, except for the sole use of the purchaser of this volume, without the prior, written permission of the copyright holder. No warranty in respect of the contents of this volume, nor to their suitability for any purpose, is expressed or implied.

Any enquiries regarding this book should be directed by mail to the publisher.

INTERFACE PUBLICATIONS also publishes:

*GETTING ACQUAINTED WITH YOUR VIC 20* by  
Tim Hartnell \_ £6.95

*ZAP POW BOOM 30 ARCADE GAMES FOR  
THE VIC 20* by Mark Ramshaw \_ £7.95

Both these books are available from most book and computer shops, or direct by mail (no charge for p. & p.) from the publisher.

Printed in England by Commercial Colour Press, London E.7.

# Contents

Introduction .....	7
ZOMBIE ISLAND .....	9
LODESTAR/3-D MAZE .....	11
ROULETTE .....	13
SPACETREK .....	18
REACTION .....	21
QUICK (DUCK SHOOT) .....	22
• BILLBOARD .....	23
MAGIC SQUARE .....	24
SIMON .....	25
MONTE CARLO MOLECULE .....	26
BIORHYTHMS .....	27
CANNIBAL CHARLIE .....	28
OTHELLO/REVERSI .....	30
DR WATSON & DR AUDIO .....	33
ZODIAC FORTUNE TELLER .....	34
• TRANQUILLITY BASE .....	36
VINCENT VAN VIC .....	37
CRAPS .....	38
• SPEEDWAY .....	41
TREACLE BALLS .....	44
ENGULF .....	46
TIGHT SQUEEZE .....	47
FRUIT MACHINE .....	48
REVERSE .....	50
SEPULCHRE ORGAN .....	51
TRAFFIC JAM .....	52
LONG JOHN SILVER .....	55
BREAKOUT .....	56
NIGHT ON A WILD MOUNTAIN .....	57
• SPACE FIGHTER .....	58
NOUGHTS AND CROSSES .....	59
MASTERMIND .....	61
COLOURMIND .....	62
VICTIM .....	64
MOTORCYCLE JUMP .....	65
MONZA .....	67
MENTO .....	70
TIME GUARDIAN .....	71
SUPERBOWL .....	73
GOMOKU .....	75
FAIRWAY .....	77
• ROBOT NIM .....	79
ZAUPER ATTACK .....	80
VIC-ET-UN .....	83
EVOLUTION .....	84

<b>SUPERPOET</b>	<b>86</b>
<b>SYMPHONY FOR A MELANCHOLY COMPUTER</b>	<b>87</b>
<b>WITH ZEN SHOW</b>	<b>88</b>
<b>VIC CHECKERS</b>	<b>89</b>
<b>JELLY BEAN SPACE SWARM</b>	<b>92</b>
<b>HANGMAN 1 &amp; 11</b>	<b>94</b>
<b>ANTI HANG</b>	<b>96</b>
<b>FLIP</b>	<b>97</b>
<b>BATTLE</b>	<b>99</b>
<b>CAVEMAN</b>	<b>101</b>
<b>APPENDICES:</b>	
<b>ERROR MESSAGES</b>	<b>104</b>
<b>BORDER/SCREEN COLOURS</b>	<b>105</b>
<b>SPEAKER COMMANDS</b>	<b>105</b>
<b>COLOUR CODES MEMORY MAP</b>	<b>106</b>
<b>SCREEN CHARACTER CODES MAP</b>	<b>107</b>



# INTRODUCTION

The VIC 20 is an exciting computer, designed to ensure you can easily make the best use of the computer's colour and sound potential.

This book has been great fun to write, because the VIC 'co-operates' so well with the games programmer. The BASIC is fast and unambiguous, the range of noises and colour effects is practically infinite, and – despite the slight limitations of the large graphics – allows some splendid screen displays to be created, especially with the POKE command.

Although I hope you'll enter these programs as we've listed them here, and enjoy playing them, the real value of a book like this lies in what you do with (a) the programs to adapt them to make them your own; and (b) the ideas you get from them to use in new programs of your own.

I've tried to choose a wide range of programs, to show how flexible the VIC can be, and to give you some ideas of the kind of programs you can write, and the easiest way to achieve the effects you desire.

The printer reproduces all the VIC graphics, and to make the listings as clear as possible, we've printed them just as they rolled off the printer. Despite this, some of the colour commands may be a little hard to fathom out. Here's a list of the colour commands, along with REVERSE ON and REVERSE OFF, as reproduced by the printer. This should help in reading the listings.

```
10 REM "■" BLACK
15 REM
20 REM "□" WHITE
25 REM
30 REM "■" RED
35 REM
40 REM "■" CYAN
45 REM
50 REM "■" PURPLE
```

```
60 REM "■" GREEN
65 REM
70 REM "■" BLUE
75 REM
80 REM "■" YELLOW
85 REM
90 REM "■" REVERSE ON
95 REM
100 REM "■" REVERSE OFF
```

As well as using the commands as listed, several programs have REM statements explaining which colours have been used. If you can't work out what a particular colour command is, use anything you like. And by all means, change the colours to suit yourself once you've seen what a particular program is meant to be doing.

The book follows a straightforward plan. The program is introduced, and with some of the programs, information on how it was written – and how it achieves the effects it does – is included.

Well, I guess you're anxious to get down to the serious business of games-playing right now, so I'll end this introduction and let you get on with it. Just before I do, I'd like to thank the friendly team at Adda Home Computers Ltd., 154 Victoria Road, London, W3, who have helped me greatly.

Good games-playing.

TIM HARTNELL  
London, November, 1982

# ZOMBIE ISLAND

Your plane has crashed on a remote island, where the only topological features are round green swamps, and the only inhabitants mindless zombies, shown on the display as letter Z's. The zombies are out to get you, a black 'clubs' symbol. You can move anywhere on the island, except through a swamp. The zombies are pretty stupid. They can see you, but not the swamps, and – as is well known among zombiologists – zombies drown instantly in swamps. You survive only if you manage to lure the zombies into the swamps.

You control your movements on the island with four key – "A" – up, "Z" – down, ",", – left and "." – right. Although you may find it a little strange using these keys at first, you'll soon find you are automatically pressing the right key for the direction in which you want to move. It will help you, in the early stages of using these keys, to note that the 'greater than' and 'less than' symbols point in the direction you will move if you press that key. These four keys are used in a number of programs in this book, and you may well want to use them in your own moving graphics games.

The number of zombies on the island is determined randomly (line 2055) and neither they, nor you, can move outside the area bounded in purple on the screen. The number of drowned zombies is shown near the bottom of the screen, as is the number left alive. The zombies make quaint noises as they move about, and drown with a dramatic aural flourish. The zombies move in a reasonably intelligent way, but have some preprogrammed stupidity to give you a chance to survive. The fewer zombies there are left, the faster they move, so the last one alive can give you a run for your money. Each round of the game ends with an echo of the noise your plane made when crashing. After a short pause, a new game will start automatically.

```
10 REM ZOMBIE ISLAND
20 GOSUB2000
30 GETS$:IFS$="" THEN S$=T$
40 AE=A:BE=B
50 IFS$="A" THEN A=A-22:B=B-22
60 IFS$="Z" THEN A=A+22:B=B+22
70 IFS$="," THEN A=A-1:B=B+1
80 IFS$="." THEN A=A+1:B=B-1
90 W=PEEK(A)
100 IF W=127 THEN A=AE:B=BE
110 IF W=26 THEN A=AE:B=BE
120 IF W=81 THEN A=AE:B=BE
130 POKEAE,1:POKEBE,1
140 POKEA,88:POKEB,0
150 T$=S$
160 REM CONTROL ZOMBIES
170 FORR=1TOY
180 U=0
```





```

185 IFPEEK(Q(R))=81THEN260
190 IFQ(R)>ATHENU=-1:IFRND(0)>.5THENU=-22
200 IFQ(R)<ATHENU=1:IFRND(0)>.5THENU=22
210 POKEQ(R),46:POKEP(R),1
220 Q(R)=Q(R)+U:P(R)=P(R)+U
225 IFPEEK(Q(R))=88THEN400
230 IFPEEK(Q(R))=81THENTC=TC+1:FORI=1TO20:POKE36875,255-3*I:NEXT:POKE36875,0:GOTO260
235 IFPEEK(Q(R))=127THENQ(R)=Q(R)-U:P(R)=P(R)-U
240 POKEQ(R),26:POKEP(R),2
260 IFTC=YTHEN310
270 POKE36874,250-RND(0)*15
290 NEXT
295 PRINT"SURVIVED UNHARMED"TC"OUT OF"Y
300 GOTO30
310 PRINT"SURVIVED UNHARMED"TC"OUT OF"Y
312 POKE36878,0
315 PRINT"W'YOU'VE SURVIVED!!"
320 GOTO40?
400 POKEQ(R),26:POKEP(R),2
402 POKE36874,0
405 PRINT"SURVIVED UNHARMED"TC"OUT OF"Y
406 PRINT"WHAT'S THE END!"
407 FORY=1TO100
408 POKE36877,130+RND(0)*100
420 NEXT
430 POKE36877,0
440 FORY=1TO3000:NEXT
450 RUN
2000 REM INITIALISE
2005 PRINT"C":Y=0
2010 FORJ=1TO15
2020 POKE7679+J,127:POKE38399+J,4
2030 POKE7987+J,127:POKE38707+J,4
2040 NEXT
2050 FORJ=7680TO7988STEP22:POKEJ,127:POKE30720+J,4
2054 Z=INT(RND(0)*11)+3:Z1=INT(RND(0)*11)+3
2055 IFJ>7680ANDJ<7988THENPOKEJ+Z,81:POKEJ+Z+30720,3
2056 T=0:IFJ>7680ANDJ<7988ANDRND(0)>.6THENT=1
2057 IFT=1THENY=Y+1:Q(Y)=J+Z1:P(Y)=J+Z1+30720:POKEQ(Y),26:POKEP(Y),2
2060 POKEJ+15,127:POKE30735+J,4
2070 NEXT
2080 A=7726:B=38446:REM START POSITION PLAYER
2090 POKEA,88:POKEB,0
2100 T$=""
2120 POKE36878,15
2500 RETURN

```

# LODESTAR/3-D MAZE

You are searching within a maze for the precious lodestar. At each point in the maze, you'll be told which directions from your current position are open. The display takes the form:

MOVE NUMBER 1

NORTH: OPEN

SOUTH: WALL

EAST: OPEN

WEST: WALL

LODESTAR INDICATOR READS 1200

DIRECTION?

North South East West Help

You enter the initial letter of the move you wish to make. H (for help) gets you a view of the maze from above, with your position shown as a blue asterisk. But although you can see where you are within the maze, the lodestar is not marked.

You will be pleased to see how quickly you start really visualizing the maze in three dimensions, and how you will be moving confidently through it. The maze changes from game to game, and the lodestar can be in one of three positions within it. Your score at the end depends on how few moves it took you to find the lodestar. Asking for help costs you a penalty of 15 moves! As you become familiar with the game, you'll learn how to make the best use of the feedback from the Lodestar Indicator.

If you have extra memory, and you feel particularly ambitious, you could add a '3-D display', to show the view ahead of you after each move. Lines 550 to 580 determine which of the three positions will be used for the lodestar in the game.

```
10 REM LODESTAR/3-D MAZE
20 GOSUB530
30 GOSUB370
40 M=M+1
50 PRINT" ";TAB(5);"MOVE NUMBER" M
70 PRINT"  NORTH: ";
80 IF A(D+1,E)=S THENPRINT"  OPEN"
90 IF A(D+1,E)=X THENPRINT"WALL"
100 PRINT"  SOUTH: ";
110 IF A(D-1,E)=S THENPRINT"  OPEN"
120 IF A(D-1,E)=X THEN PRINT"WALL"
130 PRINT"  EAST: ";
140 IF A(D,E+1)=S THEN PRINT"  OPEN"
150 IF A(D,E+1)=X THEN PRINT "WALL"
```

```

160 PRINT"QUEST: ";
170 IF A(D,E-1)=S THEN PRINT"OPEN"
180 IF A(D,E-1)=X THEN PRINT"WALL"
190 PRINT"LODESTAR INDICATOR":PRINT TAB(6);"READS"100*(ABS(Z-D)+ABS(Y-E))+Y-E
210 PRINT"*****DIRECTION?*"
211 PRINT"NNORTH SSOUTH SEAST SWEST SHHELP"
215 GETA$:IFA$=""THEN215
220 IFA$="N"AND A(D+1,E)=XTHEN 215
230 IFA$="S"AND A(D-1,E)=XTHEN 215
240 IFA$="E"AND A(D,E+1)=XTHEN215
250 IFA$="W"AND A(D,E-1)=XTHEN215
260 IF A$="H" THEN GOSUB370
270 IFA$="N" THEN D=D+1
280 IF A$="S" THEN D=D-1
290 IF A$="E" THEN E=E+1
300 IF A$="W" THEN E=E-1
310 IF Z=D AND Y=E THEN 340
330 GOTO 40
340 PRINT"YOU FOUND IT IN"
350 PRINTTAB(5);"NM"MOVES!!!":FORJ=1TO100:NEXTJ:GOTO340
370 REM HELP
400 PRINT"*****NORTH*":PRINT
410 FORB=1TO1 STEP-1:FORC=1TO15
420 FORC=1TO15
430 IF A(B,C)=X THEN PRINT"X";
435 IFB=D AND C=E THEN PRINT "X*":GOTO440
436 IF A(B,C)=S THEN PRINT"#+";
440 NEXT:PRINT:NEXT
470 PRINT"*****SOUTH*":PRINT
480 M=M+15
490 FORJ=1TO2000:NEXTJ
500 PRINT"Q":A(D,E)=S
520 RETURN
530 DIMA(15,15)
550 B=INT(RND(1)*3)+1
560 Z=14:Y=14
570 IF B=2 THEN Y=2
580 IFB=3 THEN Z=2
590 X=1:S=2
610 FORB=1TO15:FORC=1TO15
630 A(B,C)=X:IF RND(1)>.9THENA(B,C)=S
640 IF C<2ORC>14ORB<2ORB>14 THENA(B,C)=X
650 NEXT:NEXT
670 D=2:E=2
690 FORF=1TO68
700 READB:READC
710 A(B,C)=S
720 NEXT
730 M=-15:POKE36879,25
740 RETURN
750 DATA 2,2,2,3,2,4,2,5,2,6,2,7
760 DATA3,7,4,7,5,7,5,6,5,5,5,4,5,3,6,3
770 DATA7,3,7,4,7,5,7,6,7,7,7,8,7,9,9,8
780 DATA9,9,10,8,10,7,10,6,10,5,10,4,8,8
790 DATA10,3,11,3,12,3,13,3,14,3,14,2,7,10
800 DATA6,10,5,10,4,10,3,10,2,10,2,11,2,12
810 DATA2,13,2,14,6,11,6,12,6,13,6,14,7,12
820 DATA14,12,8,12,8,14,9,12,9,13,9,14,10,12
830 DATA11,9,11,10,11,11,11,12,12,9,13,9,13,10
840 DATA13,11,13,12,13,13,13,14,14,14

```



# ROULETTE

Roulette is one of the classic gambling games, and with this program you're ready to try out your 'system' for beating the wheel. . . without losing your shirt.



The first program allows full casino betting, playing by the European rules, which have a wheel with the numbers one to 36, plus zero. The American wheel has the numbers one to 36, plus zero and double zero. This is catered for by the second program.

The roulette table has the numbers one to 36, some in black and some in red on it, plus the zero (and, in the case of the American system, the double zero). You can place bets on a wide variety of number combinations. Once the bets are placed, a wheel is spun, and the croupier throws a ball into the wheel, in the opposite direction to the spin. The ball ends up resting on one of the numbers, and this determines your fate.

When you run the program, you'll see the words YOU HAVE 100 CHIPS. After a short pause, the words MESSIEURS, FAITES VOS JEUX (Gentlemen, place your bets) appear. You then enter a letter of the alphabet from A to P (or Q to quit the game) which indicates which type of bet you wish to place. Here is the key to enter your bets:

- A – A single number
- B – Two adjoining numbers
- C – Three numbers in adjoining columns
- D – Six numbers in adjoining columns
- E – Four numbers in a square
- F – Numbers one to 12 (known as the First Twelve)
- G – Numbers 13 to 24 (known as the Second Twelve)
- H – Numbers 25 to 36 (the Third Twelve)
- I – Numbers one to 18 (Low)
- J – Numbers 19 to 36 (High)
- K – 12 numbers in a horizontal row
- L – Two adjacent horizontal columns
- M – Any red number (see below for a list of red and black numbers)
- N – Any black number
- O – Any even number
- P – Any odd number
- Q – to quit the game

The red numbers are: 1, 3, 5, 7, 12, 14, 16, 18, 19, 21, 23, 25, 27, 30, 32, 34, 36

The black numbers are: 2, 4, 6, 8, 10, 11, 13, 15, 17, 20, 22, 24, 26, 28, 29, 31, 33, 35

You will not be allowed to bet more chips than you have, and you may withdraw from the wheel at any time. The bank only holds 1000 chips, so if you get a total of 1100 in hand (the bank's 1000, plus your original 100 chips), the game must end, when you'll be the one who broke the bank at Monte Carlo.

Once you've spent a bit of time gambling with this program, you may wish to try some of the 'systems' which evolved for roulette. One of the most popular is the Martingale in which you stick to the same number or group of numbers, doubling the bet each time you lose, so you bet one chip the first spin, two the next (if you lost on the first), four the next, and so on. This ensures, in theory, that any win should wipe out all previous losses, and give you a profit as well. Most systems are

systematic only in that they help you lose your money in record time. However, it is far better to try out a system with the program, than with an actual wheel. Sir Hiram Maxim, the inventor of the first automatic repeating gun (hardly an achievement likely to make him an expert on roulette wheels), is said to have said that "a bad system is better than none at all". Balance this against the advice of William Makepeace Thackeray: "You have not played yet? Do not do so; above all, avoid a martingale if you do".

```

1 REM ROULETTE - EUROPEAN
3 POKE36879,26
5 PRINT"J": GOTO20
10 FORK=1TO800:NEXT:PRINT:RETURN
20 DIMB(24):CH=100
40 PRINT"YOU HAVE"CH"CHIPS"
50 GOSUB10
60 PRINT"MESSIEURS, FAITES      VOS JEUX"
70 GETA$:IFA$=""THEN70
72 A=ASC(A$)-64:IFA<10A>17THEN70
75 IFA=17THEN4000
80 FORQ=1TO24:B(Q)=-99:NEXT
110 INPUT"HOW MANY CHIPS":N:IFN>CHTHEN110
120 CH=CH-N
130 IFA=1THENGOSUB2600
131 IFA=2THENGOSUB2640
132 IFA=3THENGOSUB2680
133 IFA=4THENGOSUB2720
134 IFA=5THENGOSUB2760
135 IFA=6THENGOSUB2800
136 IFA=7THENGOSUB2840
137 IFA=8THENGOSUB2880
138 IFA=9THENGOSUB2920
139 IFA=10THENGOSUB2960
140 IFA=11THENGOSUB3000
141 IFA=12THENGOSUB3040
142 IFA=13THENGOSUB3080
143 IFA=14THENGOSUB3120
144 IFA=15THENGOSUB3160
145 IFA=16THENGOSUB3200
150 GOSUB10
160 PRINT"WHEEL IS SPINNING":GOSUB10
170 GOSUB10:PRINT"J"
180 FORB=1TO75:C=INT(RND(0)*37):PRINT"J"000"C:FORT=1TO3*B
185 NEXT:GOSUB5000:NEXT
190 GOSUB10:PRINT"BALL HAS ENDED AT"C
240 Y=0:E=1
250 IFB(E)=CTHENY=1:GOTO270
260 IFE<24THENE=E+1:GOTO250
270 IFY=0THEN370
290 WI=00*N:CH=CH+WI
310 PRINT"CONGRATULATIONS, YOU  HAVE WON"WI"CHIPS"
330 GOSUB10
350 GOSUB400
360 GOTO40
370 PRINT"SO YOU LOSE":GOTO330
400 IFCH<1THENPRINT"YOU HAVE RUN OUT OF  CHIPS":END
410 IFCH>1100THENPRINT"YOU HAVE BROKEN THE  BANK!":END
420 RETURN
2600 INPUT"WHICH NUMBER":D:B(1)=D:OD=35:RETURN
2640 INPUT"WHICH TWO NUMBERS":B(1),B(2):OD=17:RETURN
2680 PRINT"WHICH NUMBER":INPUT"  IN LEFT COLUMN":D

```

```

2690 FORE=0T02:B(E+1)=D+E:NEXT:OD=11:RETURN
2720 INPUT"FIRST NUMBER OF SIX";D:FORE=0T05:B(E+1)=D+E:NEXT:OD=5:RETURN
2760 INPUT"1ST NUMBER IN SQUARE";D:FORE=0T03:B(E+1)=D+E:IFE=2THEND=D+1
2770 NEXT:OD=8:RETURN
2800 FORE=1T012:B(E)=E:NEXT:OD=2:RETURN
2840 FORE=1T012:B(E)=E+12:NEXT:OD=2:RETURN
2880 FORE=1T012:B(E)=E+24:NEXT:OD=2:RETURN
2920 FORE=1T018:B(E)=E:NEXT:OD=1:RETURN
2960 FORE=1T08:B(E)=E+18:NEXT:OD=1:RETURN
3000 PRINT"LOW NUMBER AT END":INPUT"OF LINE";D
3010 FORE=0T011:B(E+1)=3*E+D:NEXT:OD=2:RETURN
3040 INPUT"LOW NO. 1ST COLUMN";D1:INPUT"LOW NO. 2ND COLUMN";D2:IFABS(D1-D2)>1T
HEN3040
3050 FORE=0T011:B(E+1)=3*E+D1:B(E+13)=3*E+D2:NEXT:OD=.5:RETURN
3080 RESTORE
3090 FORE=1T018:READB(E):NEXT:OD=1:RETURN
3120 RESTORE:FORE=1T018:READZ:NEXT:FORE=1T018:READB(E):NEXT:OD=1:RETURN
3160 FORE=2T036STEP2:B(E/2)=E:NEXT:OD=1:RETURN
3200 FORE=1T035STEP2:B((E+1)/2)=E:NEXT:OD=1:RETURN
3210 DATA1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36
3220 DATA2,4,6,8,10,11,13,15,17,20,22,24,26,28,29,31,33,35
4000 PRINT"YOU ARE WITHDRAWING FROM THE GAME WITH "CH"CHIPS"
4010 END
5000 POKE36878,5
5010 POKE36877,130+B:POKE36875,130+B
5020 POKE36878,0:POKE36877,0:POKE36875,0
5030 RETURN

```

```

1 REM ROULETTE - AMERICAN
3 POKE36879,26
5 PRINT"J": GOTO20
10 FORK=1T0800:NEXT:PRINT:RETURN
20 DIMB(24):CH=100
40 PRINT"YOU HAVE"CH"CHIPS"
50 GOSUB10
60 PRINT"MESSIEURS, FAITES VOS JEUX"
70 GETA$:IFA$=""THEN70
72 A=ASC(A$)-64:IFA<10RA>17THEN70
75 IFA=17THEN4000
80 FORQ=1T024:B(Q)=-99:NEXT
110 INPUT"HOW MANY CHIPS";N:IFN>CHTHEN110
120 CH=CH-N
130 IFA=1THENGOSUB2600
131 IFA=2THENGOSUB2640
132 IFA=3THENGOSUB2680
133 IFA=4THENGOSUB2720
134 IFA=5THENGOSUB2760
135 IFA=6THENGOSUB2800
136 IFA=7THENGOSUB2840
137 IFA=8THENGOSUB2880
138 IFA=9THENGOSUB2920
139 IFA=10THENGOSUB2960
140 IFA=11THENGOSUB3000
141 IFA=12THENGOSUB3040
142 IFA=13THENGOSUB3080
143 IFA=14THENGOSUB3120
144 IFA=15THENGOSUB3160
145 IFA=16THENGOSUB3200

```

```

100 GOSUB10
160 PRINT"WHEEL IS SPINNING":GOSUB10
170 GOSUB10:PRINT"C"
180 FORB=1TO75:C=INT(RND(0)*38)-1:PRINT"0000"C:FORT=1TO3*B
182 IFC<0-1THENPRINT"00000000"C
183 IFC=-1THENPRINT"00000000DOUBLE ZERO"
184 FORT=1TO3*B
185 NEXT:GOSUB5000:NEXT
186 GOSUB10:PRINT"00000000"
187 IFC<0-1THENPRINT"BALL HAS LANDED AT"C
190 IFC=-1THENPRINT"BALL HAS LANDED ON DOUBLE ZERO"
240 Y=0:E=1
250 IFB(E)=CTHENY=1:GOTO270
260 IFC<24THENE=E+1:GOTO250
270 IFY=0THEN370
290 WI=00*W:CH=CH+WI
310 PRINT"CONGRATULATIONS, YOU HAVE WON"WI"CHIPS"
330 GOSUB10
350 GOSUB400
360 GOTO40
370 PRINT"SO YOU LOSE":GOTO330
400 IFCH<1THENPRINT"YOU HAVE RUN OUT OF CHIPS":END
410 IFCH>1100THENPRINT"YOU HAVE BROKEN THE BANK!":END
420 RETURN
2600 INPUT"WHICH NUMBER":D:B(1)=D:OD=35:RETURN
2640 INPUT"WHICH TWO NUMBERS":B(1),B(2):OD=17:RETURN
2680 PRINT"WHICH NUMBER":INPUT" IN LEFT COLUMN":D
2690 FORE=0TO2:B(E+1)=D+E:NEXT:OD=11:RETURN
2720 INPUT"FIRST NUMBER OF SIX":D:FORE=0TO5:B(E+1)=D+E:NEXT:OD=5:RETURN
2760 INPUT"1ST NUMBER IN SQUARE":D:FORE=0TO3:B(E+1)=D+E:IFE=2THEND=D+1
2770 NEXT:OD=8:RETURN
2800 FORE=1TO12:B(E)=E:NEXT:OD=2:RETURN
2840 FORE=1TO12:B(E)=E+12:NEXT:OD=2:RETURN
2880 FORE=1TO12:B(E)=E+24:NEXT:OD=2:RETURN
2920 FORE=1TO18:B(E)=E:NEXT:OD=1:RETURN
2960 FORE=1TO8:B(E)=E+18:NEXT:OD=1:RETURN
3000 PRINT"LOW NUMBER AT END":INPUT"OF LINE":D
3010 FORE=0TO11:B(E+1)=3*E+D:NEXT:OD=2:RETURN
3040 INPUT"LOW NO. 1ST COLUMN":D1:INPUT"LOW NO. 2ND COLUMN":D2:IFABS(D1-D2)>1T
HEN3040
3050 FORE=0TO11:B(E+1)=3*E+D1:B(E+13)=3*E+D2:NEXT:OD=.5:RETURN
3080 RESTORE
3090 FORE=1TO18:READB(E):NEXT:OD=1:RETURN
3120 RESTORE:FORE=1TO18:READZ:NEXT:FORE=1TO18:READB(E):NEXT:OD=1:RETURN
3160 FORE=2TO36STEP2:B(E/2)=E:NEXT:OD=1:RETURN
3200 FORE=1TO35STEP2:B((E+1)/2)=E:NEXT:OD=1:RETURN
3210 DATA1,3,5,7,9,12,14,16,18,19,21,23,25,27,30,32,34,36
3220 DATA2,4,6,8,10,11,13,15,17,20,22,24,26,28,29,31,33,35
4000 PRINT"0000YOU ARE WITHDRAWING FROM THE GAME WITH "CH"CHIPS"
4010 END
5000 POKE36878,5
5010 POKE36877,130+B:POKE36875,130+B
5020 POKE36878,0:POKE36877,0:POKE36875,0
5030 RETURN

```

# SPACETREK

You are in command of a galactic probe ship, responsible for the safety of a ten by ten sector of space, containing the Antares, Procyon, Rigel, Vega, Canopus, Altair and Sagittarius sectors of the known galaxy (see lines 8540 – 8610). This area of space contains a number of rather slow-witted aliens. They are immobile during a game, although their positions will, of course, be different from game to game.

You have a limited amount of energy, and your task in the game is to roam the galaxy, making use of your long and short range scanners when you choose to do so, to locate and then destroy the aliens. A direct hit is not always successful, and you firing on the aliens often enrages them so that they fire back at you. If they hit you, the damage you sustain is measured in energy units, decreasing the energy you have left to use. Moving, firing and using the scanners all consume energy. Long-range scanners use up more energy than short-range ones.

When you run the program, you'll see the sector space drawn out in the bottom left of the screen. At first it will be blank except for you – the red inverse dollar sign in the middle. Under the sector are the words YOU ARE IN RIGEL SECTOR AT 5 5 – ENERGY 1234 and at the top of the screen are your three options 1 – SCAN 2 – MOVE 3 – FIRE. You enter one of the three numbers.

SCAN works in one of two ways. If you enter 1 to show you wish to scan, the words SCANNER: SHORT (1) OR LONG (2) RANGE will appear. Short range looks at the squares immediately surrounding the one you are in, and reports if there is anything there. It does not tell you in which direction.

Long-range allows you to choose the direction, and gives a readout of two squares away in that direction. As I mentioned earlier, captain, long-range scanning uses up more energy than short-range scanning does. If you told the VIC you wished to scan north, it will reply SCANNER IS POSITIVE (which means an alien ship is to the north of you) or SCANNER IS NEGATIVE which means the square two to the north of you is empty.

You have to try and kill as many aliens as you can before you run out of energy, but as each action uses up energy, you must judge your moves most carefully before making them. If you fire into an empty square, it will turn into an inverse blue X so you know not to shoot into that square again, although you may move through it if you like. A direct hit gets an inverse purple asterisk in the relevant square.

You'll find that playing the game will teach you more about how to play it than these instructions will. When you enter FIRE or MORE as your choice, you'll be asked to indicate the direction you want. First the computer will ask N/S (for north or south)

and if you wish to move in one of these directions, enter N or S. Just press RETURN if you do not want to move up or down. Then you'll be asked to choose from east or west. Again, just press RETURN if you do not want to move east or west. You can combine moves to move, for example, north east, or south west, if you like.

The program is very tight on space, with just over 100 bytes left when the game is underway, so a number of error-trapping routines, which would have been useful, have been left out. Certainly you should add them if you have memory. Note that the short-range scanner cannot be used in the outer circuit (that is if either co-ordinate is 1 or 10). The long-range scanner cannot be used if either co-ordinate is 1, 2, 9 or 10.

There is nothing in the program to stop you trying to move off the galactic grid, but trying to do so will cause the program to crash. Add your own choice of name for an alien in line 9240 if you don't like mine.

If you have extra memory, some sound would certainly enliven the action. As well, you could add a routine to move the aliens slowly about and one to put the name of a crew member before each report (reports being direction of move or fire, location, energy level, information on whether you've been hit or not, and so), with the actual crewmember making the report changing from time to time.

```

20 GOSUB9000
25 GOSUB8000
40 PRINT"3ENERGY"INT(E)
41 IFE<1THEN3800
42 IFAL>0THENPRINT"ATALLY"AL
130 PRINT"3001--SCAN 2--MOVE 3--FIRE":INPUTD
156 IFD=1THENGOSUB1000
157 IFD=2THENGOSUB2000
158 IFD=3THENGOSUB3000
160 GOTO40
1000 GOSUB8000
1010 PRINT"3003SCANNER:"
1040 PRINT"3005SHORT(1) OR":PRINT"3LONG (2) RANGE":INPUTK
1080 E=E-10*K:F=0
1100 IFK=2THEN1500
1120 IFA(B+1,C)=1ORA(B+1,C+1)=1ORA(B,C+1)=1ORA(B-1,C)=1ORA(B-1,C-1)=1THENF=1:GOT
01130
1125 IFA(B,C-1)ORA(B+1,C-1)=1ORA(B-1,C+1)=1THENF=1
1130 IFF=0THENPRINT"3007NEGATIVE":GOTO1670
1140 IFF=1THENPRINT"301"/Z$ " NEAR":GOTO1670
1500 PRINT"DIRECTION N-1, S-2, E-3, W-4":INPUTH:Z=0
1560 IFN=1AND(A(B-2,C)=1)THENZ=1
1570 IFN=2AND(A(B+2,C)=1)THENZ=1
1580 IFN=3AND(A(B,C+2)=1)THENZ=1
1590 IFN=4AND(A(B,C-2)=1)THENZ=1
1640 PRINT"3011SCANNER IS "
1650 IFZ=1THENPRINT"POSITIVE"
1660 IFZ=0THENPRINT"NEGATIVE"
1670 FORT=1TO2000:NEXT:GOSUB8000:RETURN
2000 E=E-50:A(B,C)=0:B(B,C)=0
2010 A$="":B$=""

```

```

2060 INPUT"NDIRECTION (N/S)";A$
2100 IFA$="N"THENB=B-1
2120 IFA$="S"THENB=B+1
2140 PRINT"NOW AT"B" "C
2150 INPUT"#####(E/W)";B$
2160 IFB$="E"THENC=C+1
2165 IFB$="W"THENC=C-1
2170 PRINT"NOW AT"B" "C
2180 FORI=1TO1000:NEXT
2260 IFA(B,C)=1THEN5500
2270 A(B,C)=2:B(B,C)=2
2290 GOSUB8000
2300 RETURN
3000 G=B:A$=""
3005 PRINT"NDIRECTION OF FIRE":INPUT"(N/S)";A$:IFA$="N"THENG=G-1
3010 IFA$="S"THENG=G+1
3020 F=C
3030 INPUT"#####(E/W)";A$
3120 IFA$="E"THENF=F+1
3130 IFA$="W"THENF=F-1
3180 E=E-100
3190 IFA(G,F)<>1THEN3305
3200 PRINT"YOU HIT THE":PRINT"N"Z$
3220 AL=AL+1:B(G,F)=4:GOTO3410
3305 B(G,F)=3:PRINT"YOU MISSED, SIR"
3315 FORG=1TO1000:NEXT
3320 PRINT"THE "Z$" ARE":PRINT"SHOOTING BACK"
3325 FORG=1TO1000:NEXT
3340 IF RND(0)>.6THEN3400
3350 PRINT"THEY HIT US!":E=E-100*RND(0):GOTO3410
3400 PRINT"THE "Z$" MISSED"
3410 FORI=1TO1000:NEXT:GOSUB8000:RETURN
3800 PRINT"ENERGY BANKS EXHAUSTED"
3810 PRINT"YOU KILLED"AL"ALIEN":IFAL<>1THENPRINT"S"
3820 PRINT"ON THIS MISSIONYOUR COMMAND RATING IS "1000*(AL/8)
3830 GOTO3830
5500 PRINT"YOUR SHIP HAS LANDED ON AN"Z$" SHIP"
5540 PRINT"YOUR SHIP HAS EXPLODED"
5550 GOTO5550
8000 PRINT"#####";
8020 FORQ=1TO10
8045 FORP=1TO10
8060 IFB(Q,P)=0THENPRINT" ";
8070 IFB(Q,P)=2THENPRINT" ";
8080 IFB(Q,P)=3THENPRINT" ";
8090 IFB(Q,P)=4THENPRINT" ";
8160 NEXT:PRINT:NEXT
8500 Q=B*C:PRINT"YOU ARE IN"
8540 IFQ<10THENPRINT"ANTARES";
8550 IFQ>9ANDQ<20THENPRINT"PROCYON";
8560 IFQ>19ANDQ<30THENPRINT"PRIGEL";
8570 IFQ>29ANDQ<50THENPRINT"VEGA";
8580 IFQ>49ANDQ<70THENPRINT"CANOPUS";
8590 IFQ>69ANDQ<90THENPRINT"ALTAIR";
8600 IFQ>89THENPRINT"SAGITTARIUS";
8610 PRINT" SECTOR AT"B" "C
8640 RETURN
9000 DIMA(10,10),B(10,10)
9060 FORA=1TO40:X=INT(RND(0)*10)+1:Y=INT(RND(0)*10)+1:A(X,Y)=1:NEXT
9150 B=5:C=5:A(B,C)=2:B(B,C)=2:AL=0
9240 Z$="ANDROZIAN"
9340 E=1234
9350 POKE36879,25
9990 RETURN

```



# REACTION

Your reactions come under scrutiny in this program which uses double-height characters. When you run the program, you'll see the words STAND BY appear. Then the screen will clear, and after a random time the words OK NOW PRESS THE SPACE BAR will appear. You must hit the space bar immediately. If you do so within the time limit, the words HEY YOU DID IT come up on the screen, and the process will start again. However, the time available to you to react in will decrease.

The process will continue, with you being given a shorter and shorter time to react each time. Eventually you will fail, and a 'reaction rating' (LOUSY, FAIR, PRETTY GOOD, VERY GOOD, or YOU'RE THE CHAMP) will be given, depending on how short was the time of your last successful reaction.

```
10 REM REACTION
15 DIMR$(4)
20 PRINT"J"
30 GOSUB9000
40 A$="XXXXXXXXXXSTANDBY"
50 B$="XXXXXXXXXXOKNOWPRESSTHEXXXXXXXXXSPACEBAR"
60 D$="XXXXXXXXXXHEY,YOUDIDIT"
70 Q=50
980 PRINT"J" A$
985 FOR T=1 TO 500: NEXT
990 FOR T=1 TO 100+RND(0)*400
992 IFRND(0)>.99 THEN PRINT"J"
995 NEXT
1000 GETF$: IFF#<" " THEN 1000
1010 PRINT"J" B$
1020 E=TI
1030 GETF$: IFF#="" THEN 1030
1040 G=TI-E: PRINT"J"
1050 IF G<Q+1 THEN PRINT D$
1060 IF G>Q THEN 1120
1070 FOR T=1 TO 2000: NEXT
1075 Q=Q-INT(RND(0)*5+2)
1080 GETF$: IFF#<" " THEN 1080
1090 GOTO 980
1120 PRINT"J"
1125 IF Q>40 THEN PRINT"XXXXXLOUSY"
1130 IF Q<41 AND Q>30 THEN PRINT"XXXXXFAIR"
1140 IF Q<31 AND Q>20 THEN PRINT"XXXXXPRETTYGOOD"
1150 IF Q<21 AND Q>10 THEN PRINT"XXXXXVERYGOOD"
1160 IF Q<11 THEN PRINT"XXXXXYOU'RETHECHAMP"
1180 FOR T=1 TO 1000: NEXT
1190 PRINT"J" XXXXXXXXX
1200 PRINT"NEWGAMENCOMINGUP"
1210 FOR T=1 TO 5000: NEXT
1220 GOTO 70
9000 POKE56,28:A=32776:FORB=7184 TO 7600 STEP 2
9005 POKEB,PEEK(A):POKEB+1,PEEK(A)
9007 A=A+1:NEXT
9010 POKE36879,25:POKE36869,255
9015 POKE36867,47
9020 RETURN
```

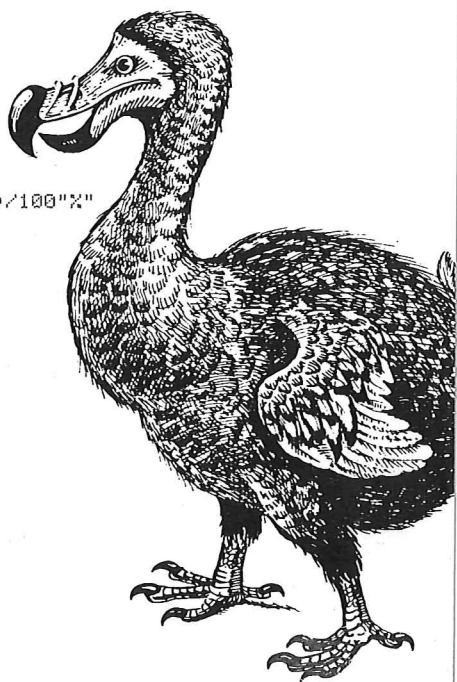


# QUACK (DUCK SHOOT)

In this rather daft game, you have to shoot the little purple ducks flying overhead. You shoot by pressing the space bar. You can move yourself right and left by pressing the "Z" and "M" keys respectively. Your score at the end of a limited time – which clocks down on the screen throughout the game – relates to the number of hits you had, compared with the number of shots you fired, so you cannot get a high score just by firing over and over again.

The quacks do not fly in a totally predictable manner, which adds to the challenge. The sound and display when you score a 'hit' is quite satisfying. Note that there is a time penalty imposed every time you press the space bar, another good reason for firing with moderation.

```
10 REM QUACK
20 PRINT"Q":POKE36879,122
30 B=7:H=10:SC=0:HI=0:FC=0
40 FORP=1TO50
45 IFP=41THENPRINT"Q"
50 PRINT"TIME"50-P" SHOTS"FC
100 FORZ=0TO20
110 POKE7768+Z,127
115 POKE7769+Z,81
120 POKE38488+Z,4
125 POKE38489+Z,4
142 GOSUB500
143 SC=SC+HI
144 IFHI=1THENPOKE36879,8:POKE36879,122
145 IFHI=1THENPRINT"SCORE"SC:HI=0
147 POKE38488+Z,B
148 POKE38489+Z,B
150 IFRND(1)>.4THENIFZ<17THENZ=Z+2
160 NEXT
170 NEXTP
180 PRINT"TIME IS UP"
190 PRINT"SUCCESS RATE"INT(10000*(SC+.00001)/(FC+.00001))/100%"
200 END
500 POKE38862+H,B
520 GETA$
530 IFA$="Z"THENIFH>1THENH=H-1
540 IFA$="M"THENIFH<20THENH=H+1
560 POKE38862+H,0
570 POKE8142+H,88
575 IFA$=" "THENGOSUB1000
580 RETURN
1000 REM SHOT
1002 FC=FC+1:P=P+1:IFP>49THENRETURN
1003 IFP=41THENPRINT"Q"
1004 PRINT"TIME"50-P" SHOTS"FC
1005 FORY=8098TO7768STEP-66
```



```

1010 POKEY+H,90
1020 POKEY+H+30720,0
1050 POKEY+H+30720,B
1060 NEXT
1065 IFH=20RH=Z+1THENHI=1:POKE36878,15:POKE36875,230+H:FOR=1T03:NEXT:POKE36878,
0
1070 RETURN

```

# BILLBOARD

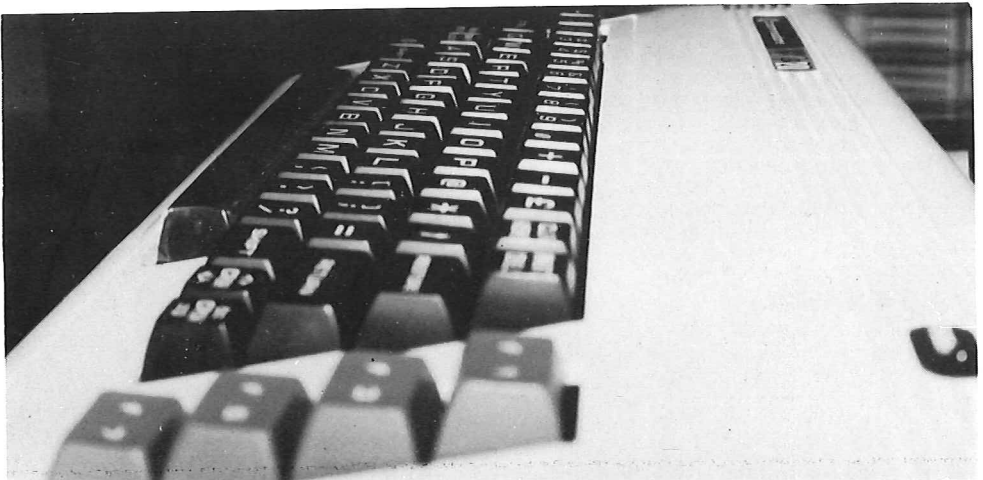
This tiny program produces very big results – twice as big, in fact. You enter the 'advertising message' you want displayed in the strings A\$, B\$ and C\$, using the 'shift right' instead of a space between the words. Only letters of the alphabet can be used with this program. Symbols, spaces and numbers are not allowed.

You may well wish to modify the program so that instead of A\$, B\$ and C\$ being permanently set, you can enter them each time you run it. Simply change line 70 into INPUT A\$, 80 into INPUT B\$ and line 90 into INPUT C\$. Note that if you do this, you can only enter single words, and you cannot use the controls to position your words.

```

10 REM BILLBOARD
70 A$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
80 B$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
90 C$="XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
100 GOSUB9000
110 PRINT"J"A$
120 FORD=1T05000:NEXT
130 PRINT"J"B$
140 FORD=1T05000:NEXT
150 PRINT"J"C$
160 FORD=1T05000:NEXT
170 GOTO110
9000 POKE56,28:A=32776:FORB=7184T07600STEP2:POKEB,PEEK(A):POKEB+1,PEEK(A):A=A+1:
NEXT
9010 POKE36879,25:POKE36869,255:POKE36867,47
9020 RETURN

```



# MAGIC SQUARE

The clever computer generates a three by three magic square, replacing three or four of the numbers with zero. As you probably know, each line of numbers in a magic square (vertically, horizontally and diagonally) adds up to the same total. It is your task to work out what the missing numbers are.

If there is already at least one complete row, it is fairly easy, with a bit of swift mental arithmetic, to determine (a) the total and (b) what digit will complete one of the other rows. This new digit should help you with another row and so on. If, however, there is not a single complete row, as sometimes happens, your task will be much harder.

The computer counts the number of moves it takes you to complete the square, and then gives you a score at the end which is related to the number of wrong guesses you made on your way to solving the problem.

```
10 REM MAGIC SQUARE
20 A=INT(RND(0)*9)+1:M=6
30 PRINT "J":J=0
40 B=INT(RND(0)*9)+1:C=INT(RND(0)*9)+1
50 IFA=BORA=CORB=CTHEN40
60 A(1)=A+B:A(2)=A-(B+C):A(3)=A+C:A(4)=A-B+C:A(5)=A
70 A(6)=A+B-C:A(7)=A-C:A(8)=A+B+C:A(9)=A-B
140 F=0
150 FORZ=1TO9:B(Z)=A(Z)
160 IFB(Z)=0THENF=1
170 NEXT:IFF=1THENRUN
180 B(A)=0
190 D=INT(RND(0)*9)+1
200 IFD<>AANDD<>BANDD<>CTHENM=M-1
205 B(D)=0
210 B(B)=0
220 B(C)=0
225 GOTO235
230 GOTO310
235 J=J+1
240 PRINT "SCORE IS GUESS NO."J
250 PRINT "J"
255 FORZ=1TO9
260 PRINTB(Z);" ";
270 IFZ=3ORZ=6THENPRINT:PRINT
280 NEXT:PRINT
290 IFM<>9THENPRINT "YOU HAVE M"RIGHT
300 IFM=9THENPRINT "YOU HAVE SOLVED IT":PRINT "YOU SCORED"INT(10000/J)
305 IFM=9THENFORT=1TO6000:NEXT:RUN
310 INPUT "YOUR GUESS";W
320 M=0
330 FORZ=1TO9
350 IFA(Z)=WTHENB(Z)=W
360 IFB(Z)<>0THENM=M+1
370 NEXT
380 GOTO235
```

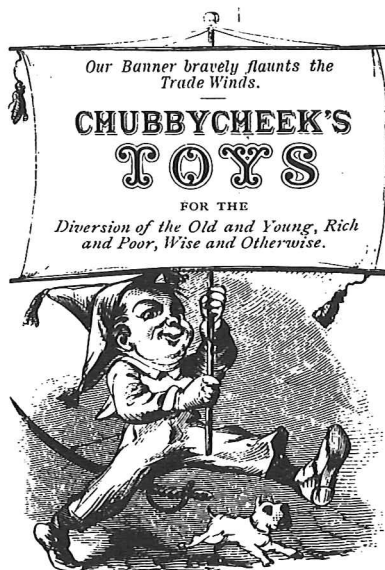
# SIMON

SIMON appears simple to play, but can be infuriatingly difficult. Your task is to copy the growing string of numbers the VIC creates. It will start with one number between one and four, printing on the screen in a distinctive colour, in a specific position, and sounding a corresponding note. You repeat the number, and it will re-appear, with the position, colour and sound as before. . . if you are correct. The first number will then be repeated, followed by a second number. You must repeat both of these, continuing as the sequence of notes and numbers builds up, until you either get a sequence of seven correct, or you make a mistake.

You'll find that you'll improve dramatically with practice. Note that if the same number appears twice in row, you have to remove your finger from the key after pressing it the first time. The border changes throughout the game to keep you on your toes.

```

5 REM SIMON
10 A$(1)="1"
20 A$(2)="2"
30 A$(3)="3"
40 A$(4)="4"
50 FORT=1T04:READS(T):NEXT
60 POKE36878,15:PRINT"J"
70 FORC=1T07
80 LETB(C)=INT(RND(0)*4)+1
90 NEXT
95 LETX=1
100 FORQ=1T0X:POKE36879,23+X
110 GOSUB700
150 NEXTQ
160 FORQ=1T0X
170 GETE$:IFE$=""THEN170
180 LETF=VAL(E$)
190 IFF<>B(Q)THEN500
200 GOSUB700
260 NEXTQ
270 X=X+1:IFX>7THEN280
275 FORT=1T0600:NEXT:GOTO100
280 PRINT"WELL DONE. YOU DID IT"
290 GOTO290
500 PRINT"YOU BLEW IT"
510 PRINT"YOU SCORED"X-1
520 END
700 C$=A$(B(Q))
705 FORT=0T01:PRINT"C"
710 FORR=1T03*(B(Q)+T)
720 PRINT"R";
730 NEXT
740 PRINTC$
745 NEXTT
750 POKE36875,S(B(Q))
760 FORY=1T0350:NEXT
762 POKE36875,0
765 PRINT"J"
770 RETURN
780 DATA195,207,215,225
    
```



# MONTE CARLO MOLECULE

The general term 'Monte Carlo Simulations' is given to those simulations which show the result of chance factors. Brownian motion and atomic diffusion are examples of random motion which can be emulated with a Monte Carlo simulation. In Monte Carlo Molecule, a molecule, starting at any position on a ten by ten grid you designate (we suggest you try 5,5 as your first one), works it way randomly around a small box until it happens to find its way to the bottom right-hand corner (co-ordinates 10,10) counting the number of moves it took to get there.

```
10 REM MONTECARLO MOLECULE
20 PRINT "Q": DIM A(10,10): M=0
25 GOSUB 1500: REM START
30 GOSUB 500: REM PRINT
35 IF P=10 AND Q=10 THEN 35
40 GOSUB 1000: REM FIND    A DIRECTION
60 GOTO 30
500 REM PRINT
505 A(P,Q)=209: M=M+1
510 PRINT "5000 MOVE NO. "M
515 PRINT "00"
520 FOR X=1 TO 10
530 FOR Y=1 TO 10
540 PRINT CHR$(A(X,Y));
550 NEXT Y: PRINT: NEXT X
560 RETURN
1000 REM DIRECTION
1010 A(P,Q)=214
1015 G=0
1020 T=INT(RND(0)*8)
1030 IF T=0 THEN GOSUB 1100
1035 IF T=4 THEN GOSUB 1120
1040 IF T=1 THEN GOSUB 1200
1045 IF T=5 THEN GOSUB 1130
1050 IF T=2 THEN GOSUB 1110
1055 IF T=6 THEN GOSUB 1140
1060 IF T=3 THEN GOSUB 1210
1065 IF T=7 THEN GOSUB 1150
1070 IF G=0 THEN 1020
1080 RETURN
1100 IF P>1 THEN P=P-1: G=1
1105 RETURN
1110 IF P<10 THEN P=P+1: G=1
1115 RETURN
1120 IF P>1 AND Q>1 THEN P=P-1: Q=Q-1: G=1
1125 RETURN
1130 IF P<10 AND Q<10 THEN P=P+1: Q=Q+1: G=1
1135 RETURN
1140 IF P<10 AND Q>1 THEN P=P+1: Q=Q-1: G=1
1145 RETURN
```

```

1150 RETURN
1200 IFQ>1THENQ=Q-1:G=1
1205 RETURN
1210 IFQ<10THENQ=Q+1:G=1
1215 RETURN
1500 REM START
1510 FORX=1TO10
1520 FORY=1TO10
1530 A(X,Y)=214
1540 NEXT NEXT
1550 PRINT"CO-ORDINATES OF"
1560 INPUT"START POSITION";P,Q
1570 IFP<1ORP>10ORQ<1ORQ>10THEN1550
1580 A(P,Q)=209
1590 PRINT"Q"
1600 RETURN

```

# BIORHYTHMS

This program is based on the three cycles which affect us throughout our lives – the physical, emotional and mental cycles. The 23-day physical cycle relates to such things as strength, aggressiveness and endurance. The emotions are said to be affected by a 28-day cycle, which relates to optimism/pessimism, frustration, temper and moodiness. The mental cycle is 33 days long, and this relates to logic, common sense, reasoning and ease of expression.

When you run the program, you'll be asked to enter your date of birth (following the clear prompts), and the day for which you want a biorhythm calculated. The computer will give you three forecasts – the day before the one you've specified, the day you want and the one which follows it.

You are then given the option for a new forecast (based on the same date of birth) or of stopping the program. A zero rating is a neutral day, positive figures are a good sign, while negative ones are warnings of a bad day in the area governed by that particular cycle. If all three are negative, you'd better stay in bed for the day.

```

10 REM BIORHYTHMS
15 REM ALASTAIR GOURLAY/TIM HARTNELL
20 PRINT"ENTER YOUR DATE OF BIRTH"
30 INPUT"YEAR (AS 1984)";A
40 INPUT"MONTH (AS 7)";B
50 INPUT"DAY (AS 21)";C
60 A=A*365.25+B*30.4+C
70 PRINT"AND NOW THE DATE FOR YOUR BIORHYTHM"
80 INPUT"YEAR (AS 1984)";B
90 INPUT"MONTH (AS 7)";C
100 INPUT"DAY (AS 21)";D
110 B=B*365.25+C*30.4+D-A
112 PRINT"Q"
115 FORF=-1TO1
116 IFF=-1THENPRINT"PREVIOUS DAY:";B=B-1
117 IFF=0THENPRINT"DAY OF FORECAST:";B=B+1
118 IFF=1THENPRINT"NEXT DAY:";B=B+1
120 C=23

```

```

130 PRINT"PHYSICAL:";
140 GOSUB250
150 C=28
160 PRINT"EMOTIONAL:";
170 GOSUB250
180 C=33
190 PRINT"INTELLECTUAL:";
200 GOSUB250
205 NEXTF
210 PRINT"ENTER 'Y' FOR A NEW FORECAST, 'N' TO STOP"
220 GET W$:IFW$=""THEN220
230 IFW$="Y"THEN?0
240 END
250 PRINTINT(100*SIN((B-INT(B/C)*C)/C*44/7));"%
260 RETURN

```

# CANNIBAL CHARLIE

You get to play God in CANNIBAL CHARLIE. You specify how many cannibals there will be at the start of creation, and how many explorers. Your aim is to create a society that survives as long as possible. Too few explorers, and the society will die within a month or so. Too many explorers, and they'll overwhelm the cannibals, again bringing things to a halt within the first month. The program gives you a population readout at the end of each month. There is a 'highest score' (i.e. longest survival of a particular society) feature, so you can try to out-God yourself. If you don't like the scenario, you can change it to foxes and rabbits, or any predator/prey combination you prefer.

The variable FD (food, set in line 100) determines how many cannibals can feast on a single explorer. Although this does not change within a single series of games, it will be different each time you run the program from the beginning.

```

10 REM CANNIBAL CHARLIE
20 PRINT"WELCOME TO CANNIBAL CHARLIE"
30 PRINT"IN THIS GAME YOU ARE GOD, AND IT IS YOUR JOB TO SET UP THE"
40 PRINT"RULES FOR CANNIBALS AND EXPLORERS IN SUCH A WAY THAT THE SOCIETY";
50 PRINT"LASTS FOR THE LONGEST POSSIBLE TIME. PRESS ANY KEY WHEN"
60 PRINT"YOU'RE READY TO PLAY"
70 GETA$:IFA$=""THEN?0
80 HI=0
100 FD=RND(0)
110 PRINT"FINE, GOD. NOW HOW MANY CANNIBALS WILL WE";
120 PRINT"START WITH (<100)?";
125 INPUTCP:IFCP>99THEN125
170 PRINT"CANNIBALS:"
180 PRINT"POPULATION:"CP
190 CP=CP/3
210 PRINT"NOW, THEN EXPLORERS:"
220 INPUT"POPULATION (<100)";EP
225 IFEF>99THEN220
230 EP=EP/3

```



```

250 PRINT"0000PLEASE STAND BY"
260 GOSUB1000:PRINT"J"
270 DA=0:CS=CS/2:ES=ES/2
280 DA=DA+1
290 PRINT"0000MONTH"DA:IFDA=1THEN300
300 IFCP>EP/FDTHENCP=EP/FD
310 CP=CP+((8*CP-CP*EP/3)*FD)
320 EP=EP+((4*EP-EP*CP)*.01)
330 PRINT"0000"INT(CP)"CANNIBALS"
340 PRINT"0000"INT(EP)"EXPLORERS"
350 IFEPC20RCP<2THEN390
360 GOSUB1000
370 GOTO280
390 IFDA>HS THEN HS=DA
395 IFCP<10REP<1THENPRINT"J"
400 PRINT"0000END OF THE LINE, GOD"
410 PRINT"0000THE SOCIETY SURVIVED FOR"DA"MONTHS"
420 PRINT"0000BEST SO FAR IS"HS
430 PRINT"0000INPUT 'Y' FOR A NEW CREATION, ANY OTHER KEY TO STOP"
440 GETA$:IFR$=""THEN440
450 IFR$="Y"THEN110
460 PRINT"0000BYE BYE, YOUR ALMIGHTYNESS"
470 END
1000 FORY=1TO4000:NEXT
1010 RETURN

```



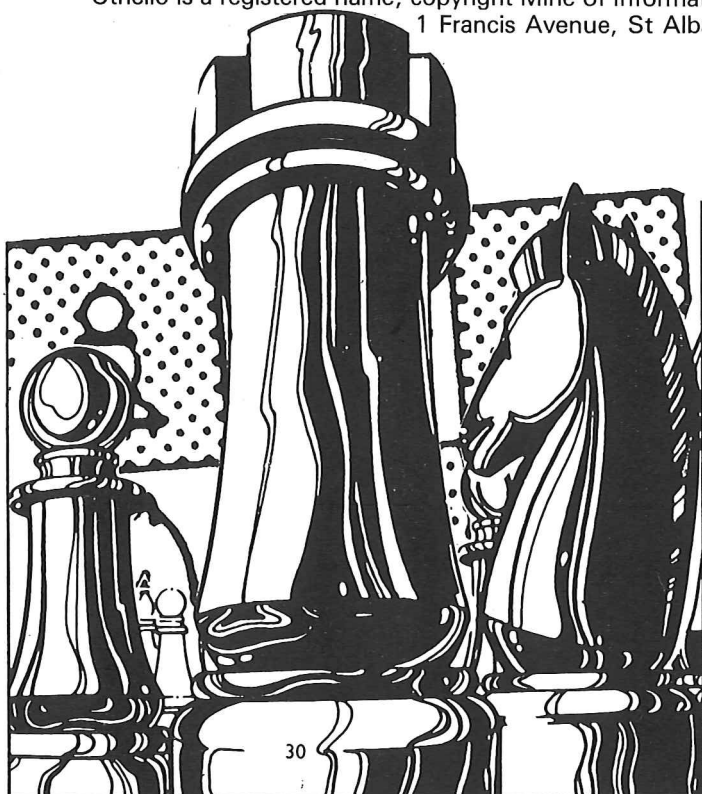
# OTHELLO/ REVERSI

Othello is based on the game Reversi which was invented in 1888. Reversi was played on a standard draughts board, using double-sided pieces – red on one side, black on the other. In his splendid book "Discovering Old Board Games" (Shire Publications Ltd., Aylesbury, 1980), R.C. Bell explains that black begins the game by placing a piece, black side up on one of the four central squares on the empty board. Red replies by placing her first piece, red side up on another central square. "These four squares are covered in the first four turns of play, and then the players continue alternately, placing their pieces on a square adjacent to one occupied by an enemy piece," Mr Bell writes.

Any enemy pieces in a straight line between the latest piece placed and another one of the player's pieces, is then turned over to show the player's colour. The winner is the player with the most pieces when the board is completely covered, or when neither player can move.

You'll find the VIC plays slowly, but remarkably well, in this version of the game. You get the choice of going first or second. You move by entering the number of the square down the side, then the number across, entering both as one double-digit number (such as 36).

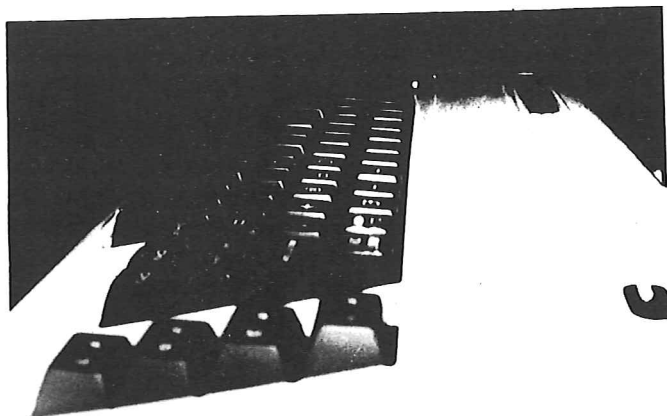
Othello is a registered name, copyright Mine of Information,  
1 Francis Avenue, St Albans.



```

1 REM OTHELLO CHARLTON/HARTNELL
2 REM NOTE LETTER O, NOT ZERO
3 REM CLS IS CLEAR SCREEN
4 PRINT"O":REM CLS
5 X=ASC("X"):O=ASC("O")
6 POKE 36879,24:REM BLACK BORDER
10 DIMA(10,10)
20 FORB=1TO10
30 FORC=1TO10
40 IF B<1 AND C<1 AND B<10 AND C<10 THEN A(B,C)=ASC(".")
50 NEXTC
60 NEXTB
70 A(5,5)=X
80 A(6,6)=X
90 A(6,5)=O
100 A(5,6)=O
110 P=0
120 PRINT"DO YOU WANT TO GO":REM CLS, RED
130 PRINT"FIRST (1=YES, 2=NO)"
140 INPUT W
150 PRINT"X":REM CLS, BLUE
160 GOSUB 3000
170 IF W=1 THEN 2000
1000 PRINT" MY MOVE
1010 S=O
1020 T=X
1030 H=0
1040 FORA=2 TO 9
1050 FORB=2 TO 9
1060 IF A(A,B)<>46 THEN 1320
1070 Q=0
1080 FORC=-1 TO 1
1090 FORD=-1 TO 1
1100 K=0
1110 F=A
1120 G=B
1130 IF(A+F+C,G+D)<>S THEN 1180
1140 K=K+1
1150 F=F+C
1160 G=G+D
1170 GOTO1130
1180 IF(A+F+C,G+D)<>T THEN 1200
1190 Q=Q+K
1200 NEXTD
1210 NEXTC
1220 IFA=2 OR A=9 THEN Q=Q*2
1230 IFB=2 OR B=9 THEN Q=Q*2
1240 IFA=3 OR A=8 THEN Q=Q/2
1250 IFB=3 OR B=8 THEN Q=Q/2
1260 IF(A=2 OR A=9) AND (B=3 OR B=8) THEN Q=Q/2
1270 IF(A=3 OR A=8) AND (B=2 OR B=9) THEN Q=Q/2
1280 IF Q<H OR Q=0 OR (PND(1)>0.3 AND Q=H) THEN 1320
1290 H=Q
1300 M=A
1310 N=B
1320 NEXTB
1330 NEXTA
1340 IFH=0 AND R=0 THEN 5000
1350 IFH=0 THEN 1370
1360 GOSUB 4000
1370 GOSUB 3000
2000 INPUT"YOUR GO":R
2010 S=X

```



```

2020 T=0:REM LETTER 0
2035 REM 0 TO PASS
2040 IF R=0 THEN 2090
2050 IF R<11 OR P>88 THEN 2000
2060 M=INT(R/10)+1
2070 N=R-10*INT(R/10)+1
2080 GOSUB 4000
2090 GOSUB 3000
2100 GOTO 1000
3000 PRINT "0":REM HOME
3010 C=0
3020 H=0
3030 PRINT:PRINT"00THELLO":PRINT
3032 REM BLACK,RVSON,RVSOFF
3035 PRINT"0123456789":REM RED,RVSON,RVSOFF,BLUE
3040 FORB=2 TO 9
3050 PRINTB-1;
3060 FORD=2TO9
3070 PRINT CHR$(A(B,D));
3080 IF A(B,D)=X THEN C=C+1
3090 IF A(B,D)=0 THEN H=H+1
3100 NEXTD
3110 PRINTB-1
3120 NEXTB
3130 PRINT"0123456789":REM RED,RVSON,RVSOFF,GREEN
3140 PRINT
3150 PRINT" I HAVE":C
3152 PRINT
3155 PRINT" YOU HAVE":H;" "
3157 PRINT
3170 RETURN
4000 FORC=-1 TO 1
4010 FORD=-1 TO 1

```



```

4020 F=M
4030 G=N
4040 IFA(F+C,G+D)<>S THEN 4080
4050 F=F+C
4060 G=G+D
4070 GOTO4040
4080 IFA(F+C,G+D)<>T THEN 4140
4090 A(F,G)=T
4100 IF M=F AND N=G THEN 4140
4110 F=F-C
4120 G=G-D
4130 GOTO4090
4140 NEXTD
4150 NEXTC
4160 RETURN
5000 IFC>H THEN PRINT"I WON":C:H
5010 IFH<C THEN PRINT"YOU WON":C:H
5020 IFH=C THEN PRINT"IT'S A DRAW!"
5030 END

```

# DR WATSON AND DR AUDIO

This is a pair of simple programs, in which the VIC thinks of a number between one and a hundred and challenges you to guess it. In 'Dr Watson' the feedback is words, and it is pretty easy to guess the number. 'Dr Audio', by contrast, uses a tone to let you know how close you are to the correct number. Once you've played about five games, you'll probably be quite pleased to see how good you become at interpreting the sounds.

## DR WATSON

```
5 REM DR WATSON
6 C=1:PRINT"J"
10 PRINT"I AM THINKING OF A      NUMBER BETWEEN ONE      AND 100"
30 A=INT(RND(1)*100)+1
40 PRINT"WHAT IS IT";
45 INPUT B
50 C=C+1
55 REM "WRONG" IN RED
60 IF B<A THEN PRINT"NO,";B;"IS WRONG"
65 IF B=A THEN 120
67 PRINT"GUESS";C;"IS ";
70 IF B<A THEN PRINT"TOO LOW";REM GREEN,BLUE
80 IF B>A THEN PRINT"TOO HIGH";REM PURPLE,BLUE
90 IF C<10 THEN 45
100 PRINT:PRINT"END OF THE GAME"
110 PRINT"I WAS THINKING OF";A:GOTO170
120 PRINT"YES, YOU ARE RIGHT!"
130 PRINT
140 PRINT"I WAS THINKING", "OF";A
150 PRINT
160 PRINT"IT TOOK YOU";C,"GUESSES"
170 FOR J=1 TO 2000: NEXT J
180 RUN
```

## DR AUDIO

```
5 REM DR AUDIO
6 C=1
7 POKE36878,7
10 PRINT"I AM THINKING OF A      NUMBER BETWEEN ONE      AND 100"
30 A=INT(RND(1)*100)+1
40 PRINT"WHAT IS IT";
45 INPUT B
50 C=C+1
```

```

55 REM "WRONG" IN RED
60 IF B<>A THEN PRINT"NO,";B;"IS WRONG"
65 IF B=A THEN 120
67 PRINT"GUESS";C;"IS ";
70 POKE 36875, ABS(A-B)+128
80 FOR T=1 TO 700: NEXT T
85 POKE 36875,0
90 IF C<30 THEN 45
100 PRINT:PRINT"END OF THE GAME"
110 PRINT" I WAS THINKING OF";A:GOTO170
120 PRINT"YES, YOU ARE RIGHT!"
130 PRINT
140 PRINT" I WAS THINKING" ,"OF";A
150 PRINT
160 PRINT"IT TOOK YOU";C,"GUESSES"
170 FOR J=1 TO 2000: NEXT J
180 RUN

```

# ZODIAC FORTUNE TELLER

ZODIAC FORTUNE TELLER, written by A.G.T. Stevens, uses a question and answer method to get the VIC to earn its keep providing sideshow entertainment. The results should not be taken too seriously.

```

5 PRINT"J"
10 GOSUB 840
20 PRINT
30 PRINT"WILL YOU PLEASE "
40 PRINT" TELL ME YOUR NAME"
50 INPUT A$
60 PRINT"J"
65 GOSUB 840
67 PRINT
70 PRINT"THANK YOU ";A$
80 PRINT
90 PRINT"WILL YOU NOW PLEASE"
100 PRINT" TELL ME YOUR BIRTHDAY"
110 PRINT
120 PRINT"GIVE ME 4 NUMBERS"
130 PRINT
140 PRINT"MONTH FIRST"
150 PRINT" 01(JAN)-12(DEC)"
160 PRINT
170 PRINT" FOLLOWED BY DAY"
180 PRINT" 01 - 31
190 INPUT B
200 PRINT"J"
210 GOSUB 840
220 PRINT
225 PRINT" YOU WERE BORN UNDER"
230 PRINT" THE SIGN OF -"
240 PRINT
250 IF B>=0101 AND B<=0120 OR B>=1223 AND B<=1231 THEN PRINT"
260 IF B>=0121 AND B<=0219 THEN PRINT"
270 IF B>=0220 AND B<=0321 THEN PRINT"
280 IF B>=0322 AND B<= 420 THEN PRINT"
290 IF B>=0421 AND B<=0521 THEN PRINT"
300 IF B>=0522 AND B<= 0621 THEN PRINT"
310 IF B>=0622 AND B<=0723 THEN PRINT"

```



CAPRICORN"

AQUARIUS"  
PISCES"  
ARIES"  
TAURUS"  
GEMINI"  
CANCER"

```

320 IF B>=0724 AND B<=0823 THEN PRINT"          LEO"
330 IF B>=0824 AND B<=0923 THEN PRINT"          VIRGO"
340 IF B>=0924 AND B<=1023 THEN PRINT"          LIBRA"
350 IF B>=1024 AND B<=1122 THEN PRINT"          SCORPIO"
360 IF B>=1123 AND B<=1222 THEN PRINT"          SAGITTARIUS"
370 PRINT
380 PRINT"SHALL I TELL"
390 PRINT"YOUR FORTUNE          ";A$;"?"
400 PRINT
410 PRINT"          PRESS Y OR N"
420 INPUT C$
430 IF C$="Y" THEN PRINT"Y"
431 IF C$="Y" THEN GOSUB 840
440 IF C$="N" THEN GOTO 829
450 PRINT
460 J=INT(10*RND(1))+1
470 IF J=1 THEN PRINT"BITCHY PALM?A CASH          WINDFALL COULD BE DUE"
480 IF J=2 THEN PRINT"BEWARE!A FINANCIAL          DEAL COULD LOSE          YOU MONEY"
490 IF J=3 THEN PRINT"AN ACQUAINTANCE MAKES          ATTRACTIVE PROPOSAL          BE CAREFUL!"
"
500 IF J=4 THEN PRINT"INVEST SPARE MONEY          WISELY"
510 IF J=5 THEN PRINT"AVOID INDECISION STICK          TO EXISTING PLANS"
520 IF J=6 THEN PRINT"YOU ARE ENTERING          A SUCCESSFUL PHASE"
530 IF J=7 THEN PRINT"IMPORTANT TO KEEP ON          GOOD TERMS WITH          EVERYBODY"
540 IF J=8 THEN PRINT"AMBITIOUS PLANS COULD          BE IN YOUR FAVOUR"
550 IF J=9 THEN PRINT"IF YOU HAVE PREVIOUSLY          FAILED-TRY AGAIN"
560 IF J=10 THEN PRINT"NEW INTERESTS COULD          BRING FRESH BENEFITS"
570 FOR T=1 TO 150:NEXT T
580 PRINT
590 K=INT(10*RND(1))+1
600 IF K=1 THEN PRINT"CERTAIN PEOPLE MAY BE          ANYTHING BUT HELPFUL"
610 IF K=2 THEN PRINT"AVOID EMOTIONAL          ENTANGLEMENTS"
620 IF K=3 THEN PRINT"AN INTERVIEW OR MEET-          -ING COULD BRING MORE          SECURITY"
630 IF K=4 THEN PRINT"SOCIAL LIFE COULD          PROVIDE MORE VARIETY"
640 IF K=5 THEN PRINT"BA BUSY MONTH AHEAD IS          INDICATED"
650 IF K=6 THEN PRINT"NOT AN ENERGETIC MONTH AHEAD BUT TRY TO VARY          YOUR ACTIVI
TIES"
660 IF K=7 THEN PRINT"MAKE SURE YOU ARE NOT          CAUGHT OFF GUARD          LATER NEXT M
ONTH"
670 IF K=8 THEN PRINT"YOU MAY NEED HELP FROM          CLOSE CONTACTS"
680 IF K=9 THEN PRINT"IFALL IN WITH OTHERS          ARRANGEMENTS"
690 IF K=10 THEN PRINT"SOME KIND OF FAMILY          ,REUNION COULD TAKE          PLACE"
700 FOR S=1 TO 150:NEXT S
710 PRINT
720 L=INT(10*RND(1))+1
730 IF L=1 THEN PRINT"PRESS ON WITH OUTDOOR          ACTIVITIES"
740 IF L=2 THEN PRINT"PLANETARY INFLUENCES          FAVOUR TRAVEL"
750 IF L=3 THEN PRINT"AN OPPORTUNITY WILL          ARISE GIVE IT CAREFUL          CONSIDERATI
ON"
760 IF L=4 THEN PRINT"THIS MONTH IS FAVOUR-          -ABLE FOR AGREEMENTS"
770 IF L=5 THEN PRINT"TESTING TIME AHEAD          BE PREPARED!"
780 IF L=6 THEN PRINT"BE FASHIONABLE BUY          A NEW OUTFIT"
790 IF L=7 THEN PRINT"TRY TO CUT DOWN ON          INESSENTIALS OTHER-          -WISE PROBL
EMS!"
800 IF L=8 THEN PRINT"SUPERIORS COULD BE          DIFFICULT BE FIRM          BUT COURTED
US"
810 IF L=9 THEN PRINT"FA FRIEND MAY HAVE          PROBLEMS TRY TO BE          HELPFUL"
820 IF L=10 THEN PRINT"OFFERS OF HELP MAY BE          REFUSED-PERSIST YOU          ARE NEEDED"
825 END
829 PRINT"Y"
830 PRINT "CALL RIGHT THEN          ";A$;" BYE BYE"
831 END

```

```

840 PRINT TAB(3);" "
850 PRINT TAB(3);" "
860 PRINT TAB(3);" "
870 PRINT TAB(3);" "
880 PRINT TAB(3);" "
890 POKE 36878,15
900 FOR L=1 TO 100
910 POKE 36876,INT(RND(1)*128)+128
920 FOR M=1 TO 10
930 NEXT M
940 NEXT L
950 POKE 36876,0
960 POKE 36878,0
970 RETURN

```

# TRANQUILLITY BASE

Here is a short moon-landing simulation which, despite its simplicity, is good fun to play and will certainly give you a challenging task.

You start off around 500 feet above the luna surface, with a limited quantity of fuel. You control the descent of your craft by entering your choice of thrust by pressing the number keys. You have fairly tight limits within which to land safely.

Running out of fuel before you reach the ground, or hitting the ground at an unacceptable speed will cause you to crash. A minus velocity means you are falling towards the luna surface. You may well start off climbing away from the moon, but its gravity will soon draw you down.

Once you've mastered the program in its present form, you may wish to add a display, or modify the program in some other way. Giving yourself less starting fuel is a good way to make the program more difficult. F% is your fuel, V% your velocity, and H% your height.

```

10 REM TRANQUILLITY BASE
20 F%=130+RND(1)*40
30 V%=-6+RND(1)*20
40 H%=500-RND(1)*10
50 PRINT "Q"
70 PRINT "*****FUEL: "F%
80 PRINT "*****VELOCITY: "V%
90 PRINT "*****HEIGHT: "H%
100 GET A%:FOR J=1 TO 800:NEXT J
110 F%=F%-A%
120 C%=A%-2
125 A%=0
130 H%=H%+V%+C%/4
140 V%=V%+C%
145 IFF%<1ANDH%<10THEN160
150 IF H%>10THEN 60
160 IFV%>-9ANDV%<8THEN 210
170 PRINT "*****CRASH! ";
180 FOR T=1 TO 300:NEXT
190 GOTO170
210 PRINT "*****GREAT LANDING, SCHIEF"
220 PRINT:PRINT"FINAL READINGS AS YOU"
225 PRINT"TOUCHED DOWN ON THE"
226 PRINT"LUNA SURFACE:"
230 PRINT:PRINT"FUEL: "F%
240 PRINT"VELOCITY: "V%

```



# VINCENT VAN VIC

You control a red diamond using "A" (up), "Z" (down), "," (left) and "." (right) to draw out a picture on the screen. You can stop the moving diamond at any time by pressing any key except for the four indicated. You choose the colour you wish the computer to use by pressing the relevant colour key. Pressing a colour change key will automatically stop the diamond moving, so it is 'drawing' in the new colour when it is started again.

The computer starts drawing in white, so it is invisible on the white background. You can therefore move the red 'cursor' diamond to wherever you like before you start drawing. You can also move it anywhere on the screen during the course of a drawing, without leaving a line. 'Drawing in white' can also be used to erase unwanted parts of a drawing.

```
10 REM VINCENT VAN VIC
20 REM MOVE RED DIAMOND WITH THE A Z , . KEYS
30 REM TO CHANGE COLOR PRESS THE NUMBER OF THE COLOUR YOU WANT
40 REM TO STOP CURSOR MOVING HIT ANY KEY EXCEPT FOR CONTROL ONES
50 PRINT "C":GOSUB9540
1000 GETA$:IFA$=""THENA$=B$
1005 IFA$>"0"ANDIFA$<"9"THENU=VAL(A$)-1
1010 E=C:T=S
1020 IFA$="," THEN S=S+1:C=C+1
1030 IFA$="," THEN S=S-1:C=C-1
1040 IFA$="A" THEN S=S-22:C=C-22
1050 IFA$="Z" THEN S=S+22:C=C+22
1060 IFPEEK(S)=102THENC=E:S=T
1070 B$=A$
1080 FORQ=1TO600:NEXT
1200 POKEE,U
1210 POKEC,2:POKES,90
1230 GOTO1000
9540 S=7954
9550 C=38674
9560 B$=""
9590 FORZ=7680TO7701:POKEZ,102:POKEZ+30720,0:NEXT
9600 FORZ=8164TO8185:POKEZ,102:POKEZ+30720,0:NEXT
9620 FORZ=7680TO8164STEP22:POKEZ,102:POKEZ+30720,0:POKEZ+21,102:POKEZ+30741,0:NE
XT
9630 U=1
9640 RETURN
```

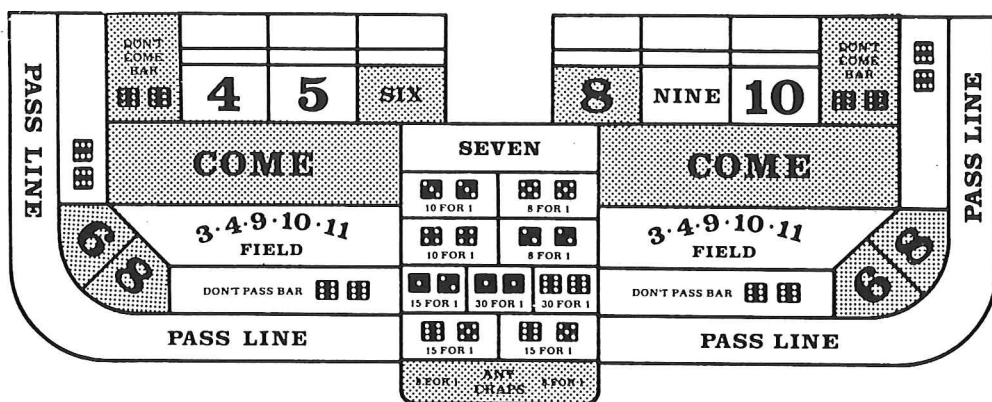
# CRAPS

Craps is a relatively simple gambling game played with two dice. The simplicity of the game itself is somewhat offset by the complexity of the betting possibilities. To make it easier to learn, we've included two versions of the game: 'Minimalist Craps' to show you the game of craps in action, and 'SF Craps' which enables you to bet following some casino-like rules.

As you can see, the first listing is very short indeed. Enter and run it, and you'll then find it much easier to understand the description of the game.

To play, you roll two dice and add up their pips. If you roll a seven or an eleven on the first roll (called a 'natural') you win, and that round is over. If you roll a two, three or twelve on the first roll (called 'craps'), you lose. Rolling four, five, six, eight, nine or ten on the first roll becomes your 'point'. The aim of the game – assuming it has not ended with the first roll – is to roll your point again, before you throw a seven. When you're sure you understand these instructions, look again at the first Craps program in action to relate what you've read here to the program.

Once you've understood craps so far, you're ready to enter the deluxe version of the game, SF Craps. In this version, you're introduced to Craps betting. You can bet on the round overall for a win ('Pass') or on the round overall for a loss ('Don't Pass'). If the game lasts more than one roll of the dice, you can have a side bet on each subsequent roll, betting that a particular number, including seven if you like, will come up. You do not need to make the side bets if you prefer to keep the game fairly simple.



```

10 REM MINIMALIST CRAPS
12 G=0:W=0:L=0
15 G=G+1
20 B$="YOU ROLLED":PRINT "J"
27 PRINT:PRINT "WINS" W, "LOSSES" L
28 PRINT:PRINT "GAME NUMBER" G
30 GOSUB 200
40 IFA=7ORA=11 THEN 240
50 IFA=2ORA=3ORA=12 THEN 230
55 P=A
60 PRINT:PRINT "YOUR POINT IS" P
70 GOSUB 200
80 IFA=P THEN 230
90 IFA=7 THEN 240
100 FOR T=1 TO 1000: NEXT
110 PRINT "J": GOTO 60
200 A=INT(RND(1)*6+RND(1)*6+1)
210 FOR T=1 TO 2000: NEXT
215 PRINT:PRINT B$A
220 RETURN
230 PRINT "YOU WIN": W=W+1: GOTO 250
240 PRINT "YOU LOSE": L=L+1
250 FOR T=1 TO 1000: NEXT
260 GOTO 15

10 REM SF CRAPS
15 PRINT "J": GOSUB 770
20 MO=20
30 GOTO 115
40 PRINT "J" "PRESS RETURN TO ROLL"
50 GET A$: IFA$="" THEN 50
60 A=INT(RND(0)*6)+INT(RND(0)*6)+2
70 GOSUB 770
80 PRINT "J" "YOU ROLLED" A
82 FOR T=1 TO 500: NEXT
85 RETURN
100 IF MO<1 THEN PRINT "YOU'RE BROKE!": GOSUB 770: GOTO 100
110 FOR T=1 TO 5000: NEXT T
115 PRINT "J": PRINT:PRINT "YOU START ROUND": PRINT TAB(8); "WITH $" MO
120 PRINT "J" "PLACE YOUR BET"
130 PRINT "J" "P" "S TO PLACE BET": PRINT "ON PASS LINE."
140 PRINT "J" "D" "S FOR DON'T PASS"
150 GET B$: IF B$="" THEN 150
155 IF B$<>"P" AND B$<>"D" THEN 150
160 B=0
170 INPUT "SIZE OF BET": B
180 IF B>MO OR B<1 THEN GOTO 170
190 GOSUB 40
200 IF A<7 AND A<11 THEN GOTO 310
210 REM NATURAL
220 PRINT "J" "YOU ROLLED A NATURAL"
230 IF B$="D" THEN 270
235 PRINT "ON PASS BET"
240 PRINT "J" "YOU WIN $" B
250 MO=MO+B
260 GOSUB 770: GOTO 110
270 PRINT "ON DON'T PASS BET"
280 PRINT "J" "YOU LOSE $" B
290 MO=MO-B
300 GOTO 100
310 IF A<2 AND A<3 AND A<12 THEN 350
320 GOSUB 770: PRINT:PRINT "SF CRAPS!"
325 FOR T=1 TO 100: NEXT T
330 IF B$="P" THEN 280
340 GOTO 240

```

```

350 PT=A
355 IF MO<1 THEN 100
360 PRINT"YOU NOW HAVE $"MO
370 PRINT"YOUR POINT IS "PT
380 PRINT"INPUT NUMBER FOR:PRINT"SIDE BET, OR PRESS"
390 PRINT"RETURN C FOR NON-BET"
400 K$="":C=0
410 INPUTK$
420 IF K$="" THEN 490
430 C=VAL(K$)
440 IF C<2 OR C>12 THEN 400
450 INPUT"HOW MUCH";D
470 GOSUB 710
480 IF D>MO OR D<1 THEN 450
490 GOSUB 60
497 IFA=7THEN630
500 IFA=PTTHENPRINT"YOU'VE MADE YOUR POINT":IFC=0THEN230
520 IFC=0THEN355
530 IF C<>A THEN 580
540 PRINT"AND YOU WIN A SIDE"
550 PRINT "BET OF $"WI
560 MO=MO+WI
565 IFA=PTTHEN230
570 GOTO355
580 PRINT"YOU LOSE $"WI
590 PRINT"ON YOUR SIDE BET"
600 MO=MO-WI
605 IF MO<1THEN 100
607 IFA=PTTHEN230
610 GOTO355
630 PRINT"YOU ROLLED 7...."
640 IFB$="D"THENPRINT"ON DON'T PASS BET":GOTO240
660 PRINT"ON PASS BET":GOTO280
710 REM ODDS/WIN
720 WI=D
730 IF A=6 OR A=8 THEN WI=INT((8*D/5)+.5)
740 IF A=5 OR A=9 THEN WI=INT((5*D/2)+.5)
750 IFA=4 OR A=9 THEN WI=2*D
760 RETURN
770 REM SOUND
780 POKE 36878,15
790 FOR T=10 TO 180 STEP RND(1)*10+.5
800 POKE 36876,255-T
820 NEXT T
830 POKE 36876,0
840 POKE 36878,0
850 RETURN

```



# SPEEDWAY

Drive your little car around the track, under control of the joystick, or keyboard ("A" – up, "Z" – down, ",", – left, "." – right) at a choice of nine different skill levels. Your score is related to both the skill level, and to the length of time you manage to keep your car going without hitting the boundaries of the track.

There is a highest score feature, so you can compete against yourself in subsequent runs. Your current score is clocked up continuously in the top right hand corner of the screen. Once you're familiar with the game in its present form, change the racetrack (from line 8040) to one of your own design, or write a routine to produce a different race track each time you play the game.

```

10 REM SPEEDWAY
15 REM JOYSTICK VERSION
20 HS=0
30 GOSUB8000
520 POKE36878,15
522 FORT=1TOSK:POKE36876,130+10*T:NEXT
525 POKE36878,0:POKE36876,0
540 IFPEEK(7680+CA+22*CD)=102THENA=1
560 POKE(7680+CA+22*CD),35
570 POKE(38400+CA+22*CD),1
580 AE=CA:ED=CD
590 GOSUB2000
600 IFX=0THENX=B
620 IFCD>2ANDX=22THENC=CD-1
630 IFCD<22ANDX=-22THENC=CD+1
640 IFCA>1ANDX=1THENCA=CA-1
650 IFCA<20ANDX=-1THENCA=CA+1
655 B=X:IFX=22THENX=-22
656 SC=SC-X:PRINT"@"SC*(10-SK)
660 POKE(38400+AE+22*ED),0
690 IFA=0THEN520
710 POKE36878,15
720 FORY=150T0250STEP5
730 POKE36877,Y
740 FORT=1TOY/5:NEXT
750 NEXT
760 POKE36878,0:POKE36877,0
765 SC=SC*(10-SK)
770 PRINT"#####YOU SCORED"SC
780 IFHS<SCTHENHS=SC
790 PRINT"#####HIGH SCORE IS"HS
800 FORT=1T02000:NEXT
810 GOT030
2000 POKEDI,127:S3=-((PEEK(PB)AND128)=0):POKEDI,255
2010 P=PEEK(PA):S1=-((PAND8)=0):S2=-((PAND16)=0):S0=-((PAND4)=0)
2020 FR=-((PAND32)=0):X=S2+S3:Y=S0+S1
2030 X=-JS(X+1,Y+1)
2040 RETURN
8000 REMINITIALISE
8005 PRINT"J"
8007 POKE36879,8
8010 A=0:REM CRASH FLAG
8012 GOSUB9000
8015 SC=0:REM SCORE
8020 CA=2:CD=3:REM STARTING POSITION OF CAR

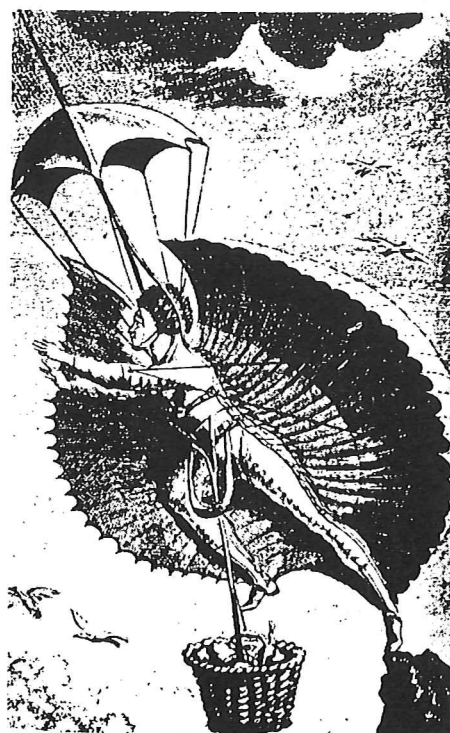
```



```

655 B#=A#
656 SC=SC+1
660 POKE(38400+AE+22*ED),1
690 RETURN
700 REMCRASH
710 POKE36878,15
720 FOR Y=150 TO 250 STEP 5
730 POKE36877,Y
740 FOR T=1 TO Y/5:NEXT
750 NEXT
760 POKE36878,0:POKE36877,0
765 SC=SC*(10-SK)
770 PRINT"SO YOU SCORED"SC
780 IF HS<SC THEN HS=SC
790 PRINT"THE HIGH SCORE IS"HS
800 FOR T=1 TO 2000:NEXT
810 GOTO 30
8000 REMINITIALISE
8005 PRINT"C"
8010 A=0:REM CRASH FLAG
8012 GOSUB 9000
8015 SC=0:REM SCORE
8020 CA=2:CD=3:REM STARTING POSITION OF CAR
8030 B#=""
8040 PRINT"THE ";
8050 PRINT"THE ";
8060 PRINT"THE ";
8070 PRINT"THE ";
8080 PRINT"THE ";
8090 PRINT"THE ";
8100 PRINT"THE ";
8110 PRINT"THE ";
8120 PRINT"THE ";
8130 PRINT"THE ";
8140 PRINT"THE ";
8150 PRINT"THE ";
8160 PRINT"THE ";
8170 PRINT"THE ";
8180 PRINT"THE ";
8190 PRINT"THE ";
8210 PRINT"THE ";
8220 PRINT"THE ";
8230 PRINT"THE ";
8240 PRINT"THE ";
8250 PRINT"THE ";
8260 PRINT"THE ";
8500 RETURN
9000 PRINT"THE ENTER YOUR SKILL"
9010 PRINT"THE LEVEL (1 TO 9)"
9020 GET B#:SK=VAL(B#)
9030 IF SK<1 OR SK>9 THEN 9020
9040 RETURN

```



# TREACLE BALLS

In this game of logic and deduction, you roll balls down a series of chutes numbered one to seven. There are three pools of treacle in the chutes. They may all be in the same chute, in three different chutes, or two in one, and the third in a different chute. They may all be in the same chute, in three different chutes, or two in one, and the third in a different chute. You have six seconds (guesses) to work out where the treacle lies.

When a ball hits treacle it either sinks (you'll be told THE BALL VANISHED) or is deflected to the chute to the right or left of the one it is in. If there is treacle in two adjoining chutes, they may both act on the ball before it reaches the end, confusing you to some extent.

If you wish to guess where the balls are, you enter '8' instead of a chute number, and then enter your three guesses. You will be told how many you had correct. If you think there are two pools in the same chute, you must enter the number twice in your guesses. If you fail to guess where the treacle lies within six seconds, you'll be told where it is, and the game ends. If you wish to make the game a little easier the first few times you play it, change the six in line 70 to a higher number.

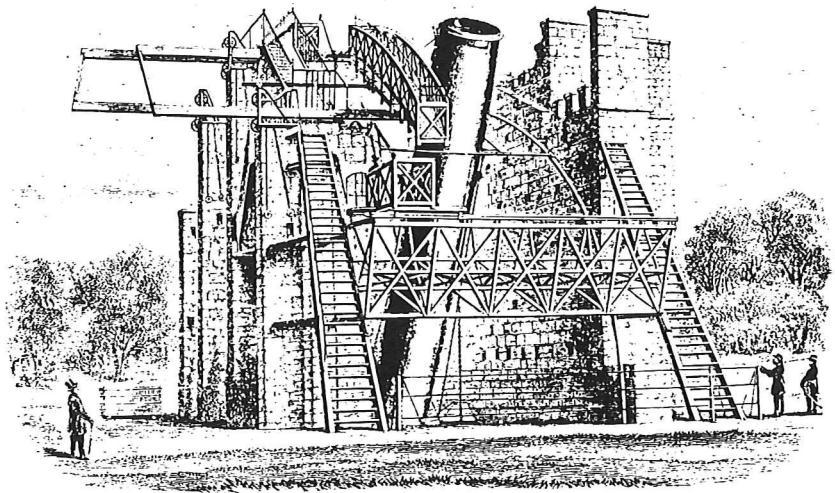
```
10 REM TREACLE BALLS
20 REM POSITION TREACLE
30 FORA=1TO3
40 B(A)=INT(RND(1)*7)+1
50 NEXT
60 REM COUNT GUESSES
70 FORH=6TO1 STEP-1
80 PRINT"YOU HAVE"H"SECONDS"
85 PRINT"ENTER BALL CHUTE NO."
86 PRINT"OR 8 IF YOU THINK YOU"
87 PRINT"KNOW WHERE THE"
88 PRINT"TREACLE LIES"
90 INPUTC:IFC=8THEN5000
95 GOSUB3000
100 D=1
110 IFB(D)=CTHEN1000
120 IF D<3 THEN D=D+1:GOTO110
125 GOSUB4000
130 PRINT"THE BALL ROLLED"
140 PRINT"OUT OF"C
150 FOR Z=1TO2000:NEXT
160 NEXTH
170 PRINT"SORRY, TIME IS UP"
180 PRINT"THE TREACLE WAS IN"
190 FOR A=1TO3
200 PRINT B(A)
210 NEXT
220 END
```



```

1000 REM TREACLE!
1010 IF RND(1)>=.9 THEN 2000
1020 IF C=1 THEN C=2:GOTO100
1030 IF C=7 THEN C=6:GOTO100
1040 F=INT(RND(1)*2)-1
1050 C=C+F
1075 GOTO100
2000 REM LOST
2010 PRINT"THE BALL VANISHED"
2020 GOTO150
3000 PRINT"XXXXXXXX"TAB(3+C):C
3015 FORZ=1TO5
3020 FORY=1TO10*Z:NEXTY
3030 PRINT TAB(5);"TTTTTTT "
3040 NEXT
3050 RETURN
4000 PRINTTAB(3+C);"X":C"XX"
4030 RETURN
5000 REM TRIAL GUESS
5010 PRINT"OK, WHERE DO YOU"
5020 PRINT"THINK THE TREACLE"
5030 PRINT"POOLS ARE?"
5040 INPUTQ(1),Q(2),Q(3)
5050 SR=0
5055 FORY=1TO3:W(Y)=B(Y):NEXTY
5060 FORN=1TO3
5070 FORM=1TO3
5080 IFQ(M)=B(N)THENSR=SR+1:B(N)=0:Q(M)=0
5090 NEXTM,N
5105 FORP=1TO3
5105 B(P)=W(P)
5107 NEXT
5110 IF SR>2THEN 5150
5120 IF SR>0 THEN PRINT "YOU GOT"SR"RIGHT"
5130 IF SR=0 THEN PRINT"WRONG, WRONG, WRONG"
5140 GOTO150
5150 PRINT"YES, YOU GOT THEM"
5160 PRINT"WITH"HS"SECONDS TO GO"

```



# ENGULF

In this game, you are moving about (most musically, as you'll hear) trying to avoid being trapped by the coloured blocks the computer is placing on the screen. You have to keep moving as long as you can. Eventually, thanks to the clever VIC, you become engulfed by blocks, and will not be able to move any more. you'll be given a score related to how long you managed to remain free.

Your movements are controlled by the same keys as in Zombie Island — "A" up, "Z" down, ",", left and "." right.

```
10 REM ENGULF
20 GOSUB9490
30 SC=0:REM SCORE
1000 GETA$:IFA$=""THENA$=B$
1010 E=C:Q=S
1020 IFA$=","THEN S=S+1:C=C+1:N=195
1030 IFA$=","THEN S=S-1:C=C-1:N=207
1040 IFA$="A"THEN S=S-22:C=C-22:N=215
1050 IFA$="Z"THEN S=S+22:C=C+22:N=225
1060 N=PEEK(S):F=0
1070 IFW=102THENC=E:S=Q
1075 IFPEEK(Q+22)=102THENF=F+.25
1076 IFPEEK(Q-22)=102THENF=F+.25
1077 IFPEEK(Q-1)=102THENF=F+.25
1078 IFPEEK(Q+1)=102THENF=F+.25
1090 IFF=1THEN2000
1100 SC=SC+1
1110 H=7680+INT(RND(0)*484):IFPEEK(H)=102THEN1110
1120 J=H+30720:POKEH,102
1130 M=INT(RND(0)*7):IFM=1THEN1130
1140 POKEJ,M
1150 POKE36874,N:POKE36875,N+1
1200 POKEE,1
1210 POKEC,2:POKE5,216
1220 B$=A$
1230 GOTO1000
2000 POKE36878,0
2010 PRINT"XXXXXXXXXXI'VE GOT YOU!!"
2020 PRINT"XXXXXXXXXXYOU SCORED"SC*236
2030 GOTO2030
9490 PRINT"J"
9540 S=7954
9550 C=38674
9570 B$=","
9590 FORZ=7680TO7701:POKEZ,102:POKEZ+30720,2:NEXT
9600 FORZ=8164TO8185:POKEZ,102:POKEZ+30720,2:NEXT
9620 FORZ=7680TO8164STEP22:POKEZ,102:POKEZ+30720,2:POKEZ+21,102:POKEZ+30741,2:NEXT
9630 POKE36878,15
9640 RETURN
```

# TIGHT SQUEEZE

This program is fun to play, especially for younger children. The VIC selects two numbers between one and 13, tells you how much money you have, and asks you to bet on the likelihood of the third number lying between the two numbers generated. If you are correct, and the third number does lie between the first two, you win double the amount you 'invested'. If you are wrong, you lose the amount. so long as you have money, you'll be offered new rounds of the game. You start with \$20, and you can bet any or all of the money you have in hand.

```

10 REM TIGHT SQUEEZE
20 D=20
25 PRINT"J":REM CLEAR SCREEN
30 A=INT(RND(1)*13)+1
40 B=INT(RND(1)*13)+1
50 IF ABS(B-A)<2 THEN 30
60 C=INT(RND(1)*13)+1
70 IFA=C OR B=C THEN 60
80 PRINT"MY FIRST NUMBER IS";A
90 PRINT"MY SECOND IS";B
100 PRINT
120 PRINT"YOU HAVE $";D;" "
125 PRINT
130 PRINT"HOW MUCH DO YOU BET"
135 PRINT"MY NEXT NUMBER LIES"
137 PRINT" BETWEEN";A;"AND";B;
150 INPUT E
160 IFE>D THEN 150
170 IFE<1 THEN PRINT"NO NUMBER!!!"
180 FORZ=1TO999:NEXTZ
190 PRINT"MY NUMBER IS";C
200 IF E<1 THEN 280
210 IF NOT (C>A AND C<B OR C<A AND C>B) THEN 250
220 PRINT"WELL DONE"
225 PRINT"YOU WIN $";2*E
230 D=D+2*E
240 GOTO 280
250 PRINT"SORRY, YOU LOSE $";E;" "
260 D=D-E
270 IF D<1 THEN 310
280 FORZ=1TO2000:NEXTZ
290 FORZ=1TO24:PRINT
292 L=INT(RND(1)*500)+1
294 C=INT(RND(1)*8)+1
296 POKE 7680 + L,160
298 POKE 38400+L,C
299 FORX=1TO50:NEXTX
300 NEXTZ
305 GOTO25
310 PRINT"THE GAME IS OVER!!!"
320 PRINT"YOU ARE BROKE ";
325 FORZ=1TO999:NEXTZ
330 GOTO320

```



# FRUIT MACHINE

This splendid program, which uses the VIC's graphics most effectively, was written for the book by M Kendall. There are three reels, and each reel has four symbols. You can stop all the reels, then either HOLD or NUDGE each one.

Due to effective government cost-cutting, you start with 5p, and each game costs you just 2p.

## The winning combinations:

Same, same, same -5p  
Same, same, any -3p  
Any, same, same -1p

## Your controls:

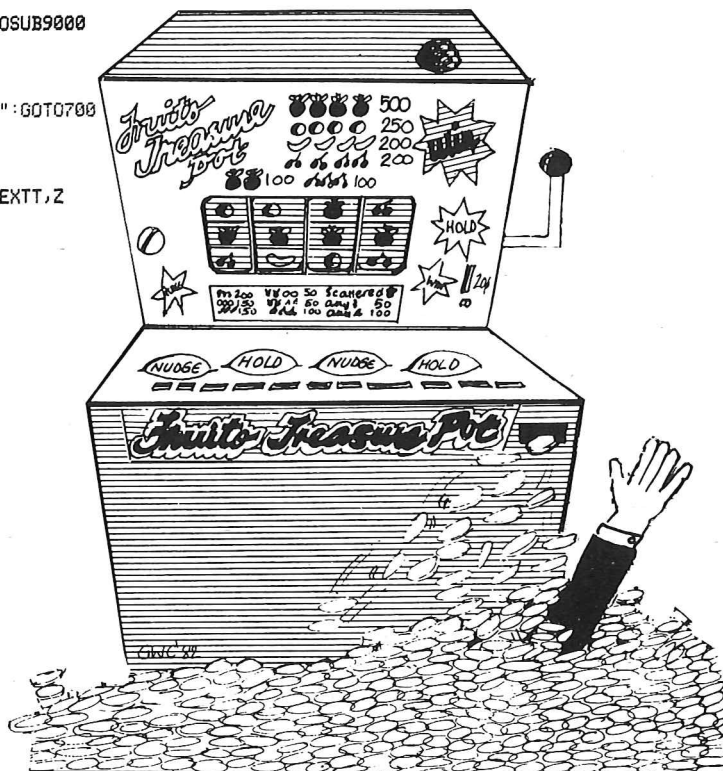
Any key to stop the reels  
N to NUDGE  
H to HOLD  
Any to play again

```
0 POKE36879,27:PRINT"BY M.KENDALL 1982"
1 PRINT"DO YOU WANT":PRINT"INSTRUCTIONS?":PRINT" (Y OR
  N)"
2 GETQ$:IFQ$=""THEN2
3 IFQ$="Y"THEN GOSUB1300
4 L=5
5 PRINT"
6 L=L-2
7 W=0
8 POKE 36878,10
9 POKE 36879,72
10 A$(1)="
11 A$(2)="
12 A$(3)="
13 A$(4)="
14 FORN=1TO3
15 A=INT(RND(1)*4)+1
16 A(N)=A
17 Q=(N-1)*7+1
18 PRINTTAB(Q)"
19 "
20 PRINT"
21 PRINTTAB(Q);A$(A)
22 POKE 36875,200:FORZ=1TO10:NEXT:POKE36875,0
23 PRINT"
24 NEXTN
25 GETQ$:IFQ$<>""THEN130
26 GOTO45
27 FORN=1TO3
28 GETQ$:IFQ$=""THEN200
29 IFQ$="H"THEN250
30 IFQ$="N"THENGOSUB300:GOTO240
31 GOTO200
32 PRINTTAB((N-1)*7+1)"NUDGE.TTTTTTT"
33 GOTO260
34 PRINTTAB((N-1)*7+1)"HOLD.TTTTTTT"
```

```

260 NEXTN
270 GOTO500
299 END
300 A=INT(RND(1)*4)+1
301 A(N)=A
305 PRINTTAB((N-1)*7+1) "      XXXXXXXX      XXXXXXXX      XXXXXXXX      XXXXXXXX
"
310 PRINT"X";TAB((N-1)*7+1)A$(A)
320 PRINT"X"
330 RETURN
500 PRINT"XXXXXXXXXXXXXXXXX=WINNINGS";
510 IFA(1)=A(2)AND A(2)=A(3)THENW=5:GOTO550
520 IFA(1)=A(2)THENW=3
530 IFA(2)=A(3)THENW=1
550 PRINTW;"IP MINUS 2P "
560 PRINT"X FOR THIS GAME"
563 IFW=5THENPRINT"X JACKPOT(T)":GOSUB9000
565 PRINT"XXXX"
570 L=L+W:PRINT"X MONEY LEFT___";L
575 IF L>20 THEN$000
580 IFL<0THENPRINT"!!! YOU ARE BROKE!!":GOTO700
600 GETQ$:IFQ$=" "THEN5
610 GOTO600
700 PRINT"XTHATS THE END FOR YOU"
701 FORZ=128TO255:POKE36877,Z:FORZ=1TO10:NEXTT,Z
702 POKE36877,0
710 GETQ$:IFQ$=" "THEN710
730 RUN
1000 PRINT"X"
1010 PRINT"X VIC FRUIT MACHINE
1020 PRINT"
1030 PRINT"X THERE ARE 3 REELSX
1040 PRINT"XAND EACH REEL HAS FOUR
1050 PRINT"XSYMBOLSX
1060 PRINT"X YOU CAN STOP ALL THE
1070 PRINT"XREELS THEN EITHER HOLD
1080 PRINT"XORNUDGE EACH ONE.X
1090 PRINT"X YOU START WITH 5P &X
1100 PRINT"XEACH GAME COSTS 2P.
1110 PRINT"XPRESS BAR TO CONTINUE."
1120 GETA$:IFA$=" "THEN1120
2000 PRINT"X"
2019 PRINT"X THE WINNING PATTERNSX
2020 PRINT"X ARE:X
2030 PRINT"X SAME SAME SAME=5PX
2040 PRINT"X SAME SAME ANY =3PX
2050 PRINT"X ANY SAME SAME=1PX
2060 PRINT"X THE CONTROL KEYS ARE:
2070 PRINT"X ANY=STOP REELSX
2080 PRINT"X N=NUDGEX
2090 PRINT"X H=HOLDX
2100 PRINT"X ANY=SPIN AGAINX
2110 PRINT"X PRESS BAR TO PLAY"
2120 GETA$:IFA$=" "THEN2120
2130 RETURN
7999 END
8000 PRINT"XCONGRATULATIONS":PRINT"XYOU'VE BROKEN THE BANK":PRINT"X I'M
OUT OF MONEY
8001 POKE36878,15
8002 POKE36879,24
8005 PRINT"XSORRY, THATS THE END"
8010 FORN=1TO12
8015 READA:POKE36875,A

```



```

8020 FORZ=1T0150:NEXTZ
8030 NEXTN
8040 DATA135,147,135,175,175,163,183,183,191,195,195,195,27
8100 RESTORE
8110 FORN=1T013
8115 READA:POKE36876,A
8120 FORZ=1T0150:NEXTZ
8130 NEXTN
8135 PRINT"■"
8140 POKE36875,0
8150 END
9000 POKE36876,10
9001 FORZ=1T010
9002 POKE36873,74
9010 FORM=200T0220-Z#2
9020 POKE36876,M
9030 NEXTM,Z
9035 POKE36876,0
9036 POKE36879,72
9040 RETURN
9500 REM-----
9510 REM
9520 REM M.KENDALL
9530 REM
9540 REM (c) 1982
9550 REM
9560 REM-----

```

# REVERSE

In this brain-boiler, the computer randomly mixes the numbers from zero to nine, and you have to put them in order in as few 'reverses' as you can. I'll explain what a 'reverse' is. If the computer had generated the sequence 5924861703 and entered 1 in response to the question "REVERSE NUMBER?", the sequence would be reprinted as 3071684295. Entering 4 would then produce 3075924861. That is, the digits from position four to the end would be reversed. As you can see from line 170, the sequence the computer is looking for is 0123456789. Get the numbers into that order, and the game is over. It may well take you 40 or so moves to order the digits, unless you're particularly clever – or lucky.

```

10 REM REVERSE
20 M=1:X=0:A$="" :PRINT"□"
30 FORT=0T09
40 L=INT(RND(1)*10)+48
50 Q=1
60 IFMID$(A$,Q,1)=CHR$(L)THEN40
70 IFQ<T THENQ=Q+1:GOTO60
80 A$=A$+CHR$(L)
90 NEXT
100 PRINT"X000 MOVE NO."M:PRINT"■";A$;"■"
110 INPUT"XREVERSE NUMBER";R:IFR<1ORR>9THEN110
120 B$=""
130 FORT=10TORSTEP-1
140 B$=B$+MID$(A$,T,1)
150 NEXTT
160 A$=LEFT$(A$,R-1)+B$
170 IFA$="0123456789"THEN190
180 M=M+1:GOTO100
190 PRINT"X001 YOU DID IT!!"
195 PRINT"■"A$
200 PRINT"MIT TOOK"M"MOVES"

```

# SEPULCHRE ORGAN

Sounding like a mournful choir in full cry in a haunted house, this program lets you play the bottom row of keys of the VIC as though it is an organ. The notes corresponding to the keys are:

C (Z), D (X), E (C), F (V), G (B), A (N), B (M), C' (,)

The note you've pressed will continue until you press another key. Touching any other key from those marked will stop the sound. RUN STOP will, of course, stop the program.

Here is TWINKLE, TWINKLE LITTLE STAR:

ZZ BB NN B  
VV CC XX Z  
BB VV CC X  
BB VV CC X  
ZZ BB NN B VV CC XX Z



```
10 REM SEPULCHRE ORGAN
12 PRINT"3"
15 DIMB(16),C$(16)
20 FORA=1TO16
30 READB(A):READC$(A)
40 NEXTA
50 DATA135,"Z",147,"X",159,"C",165,"V",175,"E",183,"N",191,"M",195,"",
55 DATA225,"Z",228,"X",231,"C",232,"V",235,"B",237,"N",239,"M",240,""
60 POKE36878,15
70 GETA$
80 FORA=1TO8
90 IFA$=C$(A)THENPOKE36876,B(A):POKE36875,B(A):POKE36874,B(A):POKE36876,B(A+8)
100 NEXTA
110 IFA$="."THENGOTO150
140 GOTO70
150 POKE36876,0
160 POKE36875,0
170 POKE36874,0
190 GOTO70
```

# TRAFFIC JAM

There are two versions of this program. The first uses the keyboard to control your car ("A" – up, "Z" – down, ",", – left and "." – right) while the second uses a joystick.

The aim of the game is to drive your car (a red asterisk) from the top left-hand corner of the screen to the car park (the red oval in the bottom right hand corner). You will be asked first to pick a skill level, from one to nine. One is the hardest, and nine the easiest. A number of cars (coloured diamonds) will appear on the screen, the number of cars being related to the difficulty level you chose. The carpark will flash off and on for a few seconds, and you'll appear in the top left hand corner to make your journey.

You can only score if you make it to the car park without hitting any of the cars, more and more of which appear as the game progresses. Your score is related to the skill level you choose, and the time it takes you to drive to the car park. The keyboard version only allows movement up, down, left and right while the joystick version of the game also allows diagonal moves.

```
10 REM TRAFFIC JAM
15 REM KEYBOARD VERSION
20 H$="000000":HS=0
30 PRINT"J"
40 GOSUB9540
1000 GETA$:IFA$=""THENA$=B$
1010 E=C:Q=S
1020 IFA$=","THEN S=S+1:C=C+1
1030 IFA$=","THEN S=S-1:C=C-1
1040 IFA$="A"THEN S=S-22:C=C-22
1050 IFA$="Z"THEN S=S+22:C=C+22
1070 W=PEEK(S)
1075 IFW=81THEN2040
1080 IFW=102THENC=E:S=Q
1090 IFW=90THEN2000
1200 POKEE,1:REM ERASE WITH WHITE
1210 POKEC,2:POKE S,35
1220 B$=A$
1230 R=INT(RND(0)*458)+7704:R2=R+30720
1240 W=PEEK(R)
1250 IFW=102THEN1230
1260 POKE36878,15:POKE36875,140+(R-7704)/4:POKE36878,0:POKE36875,0
1270 POKER,90:POKER2,VAL(MID$(P$,INT(RND(0)*6)+1,1))
1280 FORO=1TO2*G-1:NEXT
1500 GOTO1000
2000 REM CRASH
2010 PRINT"*****YOU HAVE CRASHED":M=0
2020 PRINT"*****TIME: "RIGHT$(TI$,2)" SECONDS"
2030 GOTO3000
2040 PRINT"*****YOU MADE IT!!":M=1
2050 POKE38881,1
2060 POKE38860,1
2070 GOTO2020
3000 TY=VAL(TI$)
```



```

3003 S=INT((50-TY)*100/G)*M
3005 IF S>HSTHENHS=S
3007 PRINT"SCORE:"S
3010 PRINT"HIGH SCORE:"HS
3020 FOR T=1 TO 2000: NEXT
3030 GOTO 300
5000 PRINT"SKILL LEVEL (1 TO 9)?"
5010 GET Z$: IF Z$="" THEN 5010
5020 G=VAL(Z$)
5030 IF G<1 OR G>9 THEN 5010
5040 PRINT"J"
5050 RETURN
9540 S=7703
9550 C=38423
9560 P$="654320"
9570 B$=","
9580 GOSUB 5000
9590 FOR Z=7680 TO 7701: POKE Z, 102: POKE Z+30720, VAL(MID$(P$, INT(RND(0)*6)+1, 1)): NEXT
9600 FOR Z=8164 TO 8185: POKE Z, 102: POKE Z+30720, VAL(MID$(P$, INT(RND(0)*6)+1, 1)): NEXT
9620 FOR Z=7680 TO 8164 STEP 22: POKE Z, 102: POKE Z+30720, 0: POKE Z+21, 102: POKE Z+30741, 0: NE
XT
9670 FOR U=1 TO 25-2*G
9680 R=INT(RND(0)*458)+7704: R2=R+30720
9685 POKE 36879, INT(RND(0)*127)+1
9690 W=PEEK(R)
9700 IF W=102 THEN 9680
9710 POKE 36878, 15: POKE 36875, 140+(R-7704)/4: POKE 36878, 0: POKE 36875, 0
9720 POKER, 90: POKER 2, VAL(MID$(P$, INT(RND(0)*6)+1, 1))
9730 NEXT
9733 POKE 36879, 26
9735 FOR L=1 TO 50
9736 POKE 8162, 81
9737 POKE 38882, 0
9738 FOR T=1 TO 10: NEXT
9739 POKE 38882, 5
9740 NEXT
9745 FOR T=1 TO 1000: NEXT
9750 TI$="000000"
9780 POKE 38882, 2
9999 RETURN

```

```

10 REM TRAFFIC JAM (JOYSTICK VERSION)
15 DIM JS(2,2)
20 HS=0
30 RESTORE: GOSUB 9490
510 POKEDD, 127: S3=-((PEEK(PB) AND 128)=0): POKEDD, 255
520 P=PEEK(PA): S1=-((P AND 8)=0): S2=((P AND 16)=0): S0=((P AND 4)=0)
530 FR=-((P AND 32)=0): X=S2+S3: Y=S0+S1
1010 E=C: Q=S
1020 S=S+JS(X+1,Y+1)
1025 C=C+JS(X+1,Y+1)
1070 W=PEEK(S)
1075 IF W=81 THEN 2040
1080 IF W=102 THEN C=E: S=Q
1090 IF W=90 THEN 2000
1200 POKEE, 1: REM ERASE WITH WHITE
1210 POKEC, 2: POKES, 35
1230 R=INT(RND(0)*458)+7704: R2=R+30720
1240 W=PEEK(R)

```

```

1250 IFW=102THEN1230
1260 POKE36878,15:POKE36875,140+(R-7704)/4:POKE36878,0:POKE36875,0
1270 POKER,90:POKER2,VAL(MID$(P$,INT(RND(0)*6)+1,1))
1280 FORO=1TO2*G-1:NEXT
1500 GOTO510
2000 REM CRASH
2010 PRINT"####YOU HAVE CRASHED":M=0
2020 GOTO2070
2040 PRINT"####YOU MADE IT!!":M=1
2050 POKE38881,1
2060 POKE38860,1
2070 PRINT"####TIME: "RIGHT$(TI$,2)" SECONDS"
2080 S=INT((50-VAL(TI$))*12345/6)*M
2090 IF$>HSTHENHS=S
2100 PRINT"####SCORE:"$
2105 PRINT"####HIGHSCORE:"HS
2110 FORT=1TO3000:NEXT
2120 GOTO30
5000 PRINT"####SKILL LEVEL (1 TO 9)?"
5010 GET Z$:IFZ$=""THEN5010
5020 G=VAL(Z$)
5030 IFG<1ORG>9THEN5010
5040 PRINT"J"
5050 RETURN
9490 PRINT"J"
9500 REMINITIALISE
9510 POKE37139,0:DD=37154:PA=37137:PB=37152
9520 FORZ=0TO2:FORJ=0TO2:READJS(J,Z):NEXTJ,Z
9530 DATA-23,-22,-21,-1,0,1,21,22,23
9540 S=7703
9550 C=38423
9560 P$="654320"

9580 GOSUB5000
9590 FORZ=7680TO7701:POKEZ,102:POKEZ+30720,VAL(MID$(P$,INT(RND(0)*6)+1,1)):NEXT
9600 FORZ=8164TO8185:POKEZ,102:POKEZ+30720,VAL(MID$(P$,INT(RND(0)*6)+1,1)):NEXT
9620 FORZ=7680TO8164STEP22:POKEZ,102:POKEZ+30720,0:POKEZ+21,102:POKEZ+30741,0:NE
XT
9670 FORU=1TO25-2*G
9680 R=INT(RND(0)*458)+7704:R2=R+30720
9685 POKE36879,INT(RND(0)*127)+1
9690 W=PEEK(R)
9700 IFW=102THEN9680
9710 POKE36878,15:POKE36875,140+(R-7704)/4:POKE36878,0:POKE36875,0
9720 POKER,90:POKER2,VAL(MID$(P$,INT(RND(0)*6)+1,1))
9730 NEXT
9733 POKE36879,26
9735 FORL=1TO50
9736 POKE8162,81
9737 POKE38882,0
9738 FORT=1TO10:NEXT
9739 POKE38882,5
9740 NEXT
9745 FORT=1TO1000:NEXT
9750 TI$="000000"
9780 POKE38882,2
9999 RETURN

```

# LONG JOHN SILVER

This is very simple program which you can use as the core of any program when you want to hide something on a grid. In this case the grid is 10 x 10, and after each guess you are given feedback as to where the gold is hidden.

## Structure of the program

Lines 10 and 20: 'Hide' the gold.

Line 30: Loop to count guesses

Line 60: Accepts player guess, strips it to two numbers to 'process' it.

Lines 130 to 190: Feedback on the accuracy of the guess.

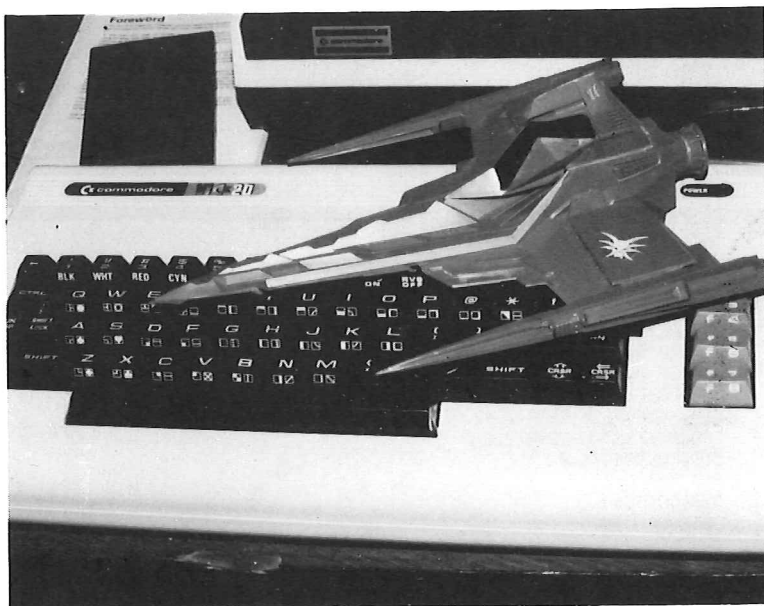
Lines 210 to 220: End of game if not guessed.

Lines 230 to 260: End if the treasure is found.

## Suggestions for improvement

- Change the size and shape (triangular?) of the grid.
- Add some ambiguity to the feedback.
- Use sound for the feedback (high notes for close, low notes for far away).
- Add a sound routine to the win routine.

```
7 POKE 36879, 15
8 PRINT "J"
10 A=INT(RND(1)*10)+1
20 B=INT(RND(1)*10)+1
30 FOR Z=1 TO 10
35 PRINT "Z":PRINT
40 PRINT "YOU HAVE ONLY"
45 PRINT 10-Z;"SECONDS LEFT!"
50 PRINT "WHERE IS THE GOLD?"
60 INPUT M:C=INT(M/10):D=M-C*10
70 IF A=C AND B=D THEN 230
130 PRINT:PRINT "IT IS NOT AT";C;D
135 PRINT
140 PRINT "HERE IS A CLUE:-"
150 PRINT "TRY TO THE ";
160 IF A<C THEN PRINT "SOUTH";
170 IF A<C THEN PRINT "NORTH";
180 IF B<D THEN PRINT "WEST";
190 IF B<D THEN PRINT "EAST";
200 NEXT Z
210 PRINT:PRINT "TIME IS UP"
215 PRINT "THE GOLD WAS AT";A;B
220 END
230 PRINT "WEEELLLLLL AAAAAA!!!"
240 S= RND(1)*10000
245 PRINT "S"
250 PRINT "YOU FOUND £";INT(S/2);
255 FOR T=1 TO 555:NEXT T
260 GOTO 250
```



# BREAKOUT

This version of one of the original computer games makes good use of the colour and music capabilities of the VIC. It is very simple to play. You have nine balls at the start of the game, and so long as you manage to bounce the ball off your slide at the bottom of the screen, the ball will stay in play.

Each time you miss the ball, your 'ball count' (top left hand corner of screen) is reduced by one. Each time the ball count is changed, your score is changed as well. You get a score every time you destroy one of the coloured bricks at the top of the screen.

You control the action of the slide with "Z" (moves it left) and "M" (moves it right).

```
1 REM BREAKOUT
5 POKE36878,15
10 GOTO40
20 X=-X
22 POKE36876,200
29 RETURN
30 Y=-Y
32 POKE36876,220
39 RETURN
40 PRINT"J"
50 SC=0:Z=9
60 FORJ=23TO100STEP 3
70 POKE 7680+J,102
75 POKE38400+J,(INT(RND(1)*8))
80 NEXT
90 F=10
110 X=-3:Y=1
120 A=20:M=A
130 B=1:N=B
160 IFA<0 THEN GOSUB 20
162 IF A>20 THEN GOSUB2000
165 A=A+X
180 IFB<1 OR B>20 THEN GOSUB 30
185 B=B+Y
200 IF PEEK(7680+22*A+B)=102 THEN GOSUB 5000
300 POKE8163+F,32
310 POKE 8165+F,32
320 POKE 8164+F,102
330 POKE38884+F,0
340 GETA$
350 IF A$="Z" THEN F=F-1
360 IF A$="M" THEN F=F+1
400 POKE 7680+22*M+N,32
500 POKE 7680+22*A+B,42
505 POKE38400+22*A+B,0
510 POKE36876,0
910 M=A:N=B
920 GOTO160
2000 GOSUB20
2005 IF ABS(F-B)<2 THEN RETURN
```

```

2010 Z=Z-1
2015 PRINT"***BALL"Z
2020 FORT=1T020
2022 POKE36876,120+ 4*T:POKE36876,0
2023 NEXT
2025 IFZ=0 THEN 6000
2030 RETURN
5000 POKE 7680+22*A+B,32
5001 POKE36876,0
5002 FORK=1T010
5003 POKE36877,255-10*K:POKE36877,0
5004 NEXT
5005 SC=SC+367
5010 PRINT"***BALL"Z" " SCORE"SC
5020 GOSUB20
5500 RETURN
6000 POKE36876,0
6005 PRINT"*** END "
6010 GOTO6010

```

# NIGHT ON A WILD MOUNTAIN

This program is perhaps best saved for those people who say: "It's very nice having a personal computer, but what can you do with it?" Once having seen (and heard) this program, they may well be convinced that perhaps they should not have asked. The music is wild and random, as befits the title, although the graphics have symmetry and order. Enter this program just as it is listed here, then try and work on it to bring some order into the music.

```

5 PRINT"J"
10 REM NIGHT ON A WILD MOUNTAIN
12 REM (RANDOM MUSIC,PATTERNS)
15 M=INT(RND(1)*8)
20 A=INT(RND(1)*18)
30 B=INT(RND(1)*18)
40 C=7911
50 D=38631
60 POKEC+A,102:POKED+A,M
70 POKEC-A,102:POKED-A,M
80 POKEC+11*A+B,102:POKED+11*A+B,M
90 POKEC-11*A-B,102:POKED-11*A-B,M
97 IF RND(1)>0.3 THEN 120
100 POKE36879,16*(INT(RND(1)*16))+INT(RND(1)*8)+8
120 POKE36878,INT(RND(1)*15)+1
130 POKE36874,INT(RND(1)*128)+128
140 POKE36875,INT(RND(1)*128)+128
150 POKE36876,INT(RND(1)*128)+128
160 POKE36877,INT(RND(1)*128)+128
195 IFRND(1)>.1 THEN205
200 POKE36878,0
205 IFRND(1)>.3 THEN215
210 POKE36874,0
215 IFRND(1)>.3 THEN230
220 POKE36875,0
230 FORT=1TORND(1)*20 +5:NEXT
235 IFRND(1)>.3 THEN250
240 POKE36877,0
250 POKE36876,0
500 GOTO10

```

# SPACE FIGHTER

You pilot your V-wing space craft across the void, trying to hit the numbers which appear at random on the screen. You must be quick, for they only stay there a short time. Your score is related to the number you run over. The sound effects in this program, and the appearance of the V-wing, are very effective, and you may well wish to use these in other space fight programs you write.

As usual, you control your ship with "A" (up), "Z" (down), "," (left) and "." (right).

```
10 REM SPACE FIGHTER
15 HS=0
20 RESTORE
25 GOSUB9490
1000 GETA$: IFA$="" THENA$=B$
1005 PRINT "SCORE" SC
1010 E=C:Q=S:V1=V
1020 IFA$="." THENS=S+1:C=C+1:V=62
1030 IFA$="," THENS=S-1:C=C-1:V=60
1040 IFA$="A" THENS=S-22:C=C-22:V=1
1050 IFA$="Z" THENS=S+22:C=C+22:V=22
1060 W=PEEK(S)
1070 IFW=102THENC=E:S=Q:V=V1
1075 F=0
1080 IFW>48THENIFW<58THENS=SC+W*791:F=1:POKE36878,15:POKE36877,0
1086 IFF=1THENFORT=135T0230:POKE36875,T:NEXT:POKE36875,0
1087 POKE36878,7:POKE36877,2*ASC(A$)+70
1090 IFRND(0)>.93THENPOKEC(CC),0:POKEB(CC),48:CC=CC+1:POKEB(CC),A(CC):POKEC(CC),
7
1100 IFCC=55THEN3000
1200 POKEE,0
1210 POKEC,7:POKEB,V
1220 B$=A$
1230 GOTO1000
3000 PRINT "TIME IS UP"
3010 PRINT "YOU SCORED" SC
3020 IFHS<SCTHENHS=SC
3030 PRINT "HIGH SCORE:" HS
3035 POKE36878,0
3040 FORT=1T03000:NEXT
3050 GOTO20
9490 PRINT "V"
9540 S=7726
9550 C=38446
9570 B$="."
9590 FORZ=7702T07723:POKEZ,102:POKEZ+30720,7:NEXT
9600 FORZ=8164T08185:POKEZ,102:POKEZ+30720,7:NEXT
9620 FORZ=7680T08164STEP22:POKEZ,102:POKEZ+30720,7:POKEZ+21,102:POKEZ+30741,7:NE
XT
9625 IFHS<0THEN9640
9630 DIMA(55),B(55),C(55)
9640 FORZ=1T055
9650 A(Z)=INT(RND(0)*9)+49
```

```

9660 B(Z)=7704+INT(RND(0)*458)
9665 C(Z)=B(Z)+30720
9670 IFPEEK(B(Z))=102THEN9660
9680 NEXT
9690 CC=1
9700 V=60
9710 SC=0:IFHS=0THENHS=1
9720 POKE36879,10
9999 RETURN

```

# NOUGHTS AND CROSSES

You are the red filled-in circles, the computer the inverse blue crosses. you just touch the number where you want to place a piece, and the VIC will do the rest. The computer will ignore any opening move which is not the centre square (5). This program is based on one written by Stuart Roberts.

```

10 REM NOUGHTS AND          CROSSES
12 REM
15 REM AFTER PROGRAM      BY STUART ROBERTS
16 REM
20 FORA=1TO9:B(A)=A:NEXT
30 PRINT"  "
50 E=0:Q=0
60 N=0
70 X=4
80 PRINT"  "
90 FORA=1TO9
100 IFA=XTHENX=X+3:PRINT:PRINT
110 IFB(A)=0THEN320
120 IFB(A)=10THEN350
130 PRINTB(A);" ";
140 NEXT
145 IFB(3)=10ANDB(6)=10ANDB(9)=10THENE=8:GOTO160
146 IFB(7)=10ANDB(8)=10ANDB(9)=10THENE=8:GOTO160
150 IFN=1THEN380
160 IFE=0THENPRINT:PRINT:PRINT"          I WIN":GOTO730
170 IFQ=5THENPRINT:PRINT:PRINT"        IT'S A DRAW":GOTO730
180 PRINT"XXXXXXXXXXYOUR MOVE?"
185 GETA$:IFA$=""THEN185
187 Z=VAL(A$)
190 IFB(Z)<>ZTHEN185
200 Q=Q+1
210 IFZ=11THENB(5)=10:GOTO60
220 N=1:B(Z)=0
230 GOTO70
240 FORA=CTOD
250 IFB(A)=ATHENB(A)=10
260 A=A+F

```

```

270 NEXT
280 GOTO60
290 IFB(5)=5THEN20
300 B(D)=10
310 GOTO60
320 PRINT"●";
330 P(A)=1
340 GOTO140
350 PRINT"■";
360 P(A)=4
370 GOTO140
380 G=0
390 C=1:D=9:F=3
400 GOSUB580
410 C=3:D=7:F=1
420 GOSUB580
430 D=9:F=2
440 GOSUB580
450 C=2:D=8
460 GOSUB580
470 C=1:D=7
480 GOSUB580
490 D=3:F=0
500 GOSUB580
510 C=4:D=6
520 GOSUB580
530 C=7:D=9
540 GOSUB580
550 G=G+1
560 IFG=5THEN60
570 GOTO390
580 E=0
590 FORA=CTOD
600 E=E+P(A):A=A+F
610 NEXTA
620 IFE=3THEN720
630 IFG=0THENRETURN
640 IFE=3THEN240
650 IFG=1THENRETURN
660 IFE=2THEN240
670 IFG=2THENRETURN
680 IFE=5THEN240
690 IFG=3THENRETURN
700 IFE=10RE=4THEN290
710 RETURN
720 PRINT:PRINT:PRINT"
730 FORT=1TO3000:NEXT
740 RUN

```

YOU WIN"





# MASTERMIND

The game Mastermind is copyright Invicta, who sell a number of versions of the game which use coloured pegs. Invicta bought the rights to the game from an amateur mathematician, Mordechai Meirovich, in 1971, and a couple of years after they first marketed it, had a worldwide best selling game.

The game has been known in England for centuries as Bulls and Cows. Another popular version is called Codebreaker.

The principle is simple. The VIC 20 picks a three-digit code (such as 259), using the number 1 to 9. Zero is not used, and no digit is used more than once in the code (so 117, 494 or 666 are impossible). you have to guess the number in 10 guesses or less. The feedback from your VIC is fairly easy to interpret. After each guess you are told YOU SCORED 2 BLACKS and 0 WHITES, in which a 'black' is a correct digit in the correct place in the three-digit code, and a white is a digit which does appear in the code, but not where you put it in your guess. The computer prints out the answer if you fail to get it within 10 guesses

```
10 PRINT"J"
30 FORZ=1TO3
40 A(Z)=INT(RND(1)*9)+1
50 NEXTZ
60 IFA(1)=A(2)OR A(1)=A(3)OR A(2)=A(3) THEN30
70 D=100*A(1)+10*A(2)+A(3)
80 FORC=1TO10
90 PRINT"GUESS ";C;
95 INPUTX
100 B(1)=INT(X/100)
110 B(2)=INT((X-100*B(1))/10)
120 B(3)=X-100*B(1)-10*B(2)
130 IFD=X THEN360
150 N=0:W=0
170 FOR E=1 TO 3
180 IFA(E) <> B(E) THEN 210
190 N=N+1
200 A(E)=0
210 NEXT E
220 FORF=1TO3
```

```

230 IFA(F)=0 THEN 280
240 FOR E=1 TO 3
250 IFB(F)O A(E) THEN 270
260 W=W+1
270 NEXTE
280 NEXTF
290 A(1)=INT(D/100)
300 A(2)=INT((D-100*A(1))/10)
310 A(3)=D-100*A(1)-10*A(2)
320 PRINT"YOU SCORED"
325 PRINT"█";N;"BLACKS &█";W;"WHITES█"
330 NEXTC
340 PRINT"MY NUMBER WAS",A(1);A(2);A(3)
350 END
360 PRINT"CONGRATULATIONS"
370 PRINT"YOU GOT IT IN";C
380 END

```

# COLOURMIND

This, as you've probably guessed, is a colour version of Mastermind. You have to guess the computer's pattern of four differently-coloured blobs. You indicate your choice of colours by pressing the colour keys on the VIC.

The colours will first print up as the numbers you pressed, then changed into blobs coloured to your choice. Following the blobs will be your score, the first number being the correct colour in the correct position, the second the correct colour in the wrong position. You'll get ten tries to guess the numbers. Note that no colour is repeated within the set of blobs.

If you're used to playing Mastermind with coloured pegs, you'll quickly appreciate that this is much closer to the Invicta game than are computer versions which use numbers. At the end of a game, press any key to get a new game, or 'S' to stop.

```

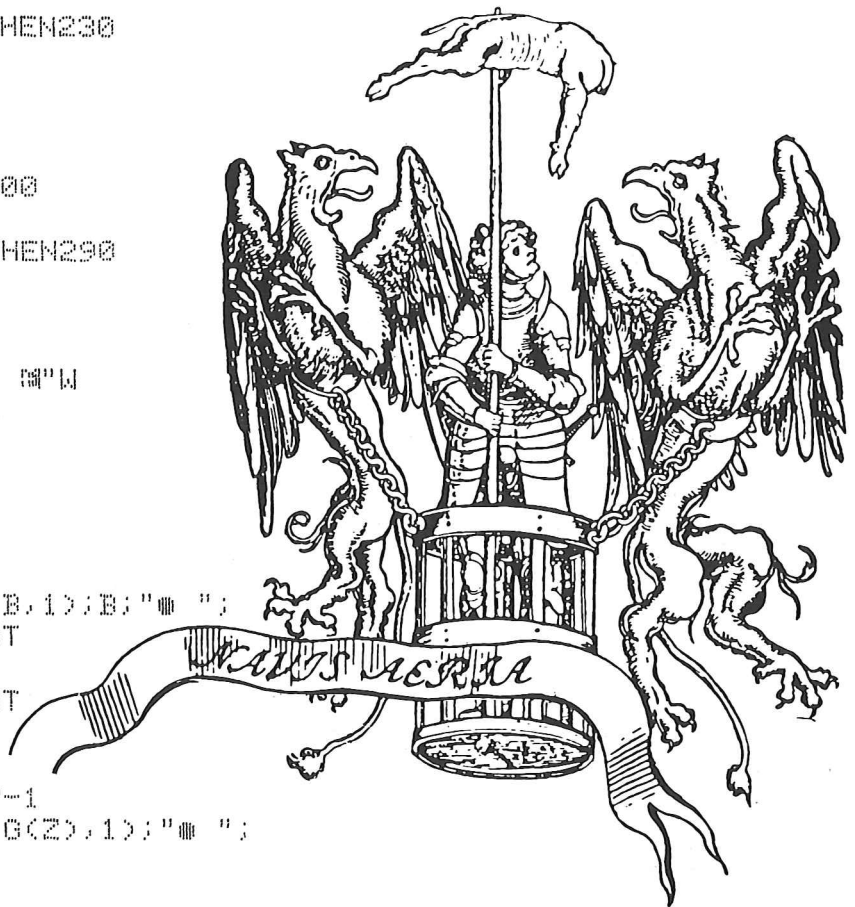
10 REM COLOURMIND
15 PRINT"J":PRINT:PRINT
20 GOSUB450
30 C(1)=INT(RND(1)*6)+3
35 Z=2
40 C(2)=INT(RND(1)*6)+3
50 J=1

```

```

60 IFC(J)=C(Z)THEN35
70 IFJ<Z-1THENJ=J+1:GOTO60
80 IFZ<4THENZ=Z+1:GOTO40
100 FORH=1TO10:INPUTA
130 FORZ=1TO4
140 G(Z)=A-10*INT(A/10)
150 A=INT(A/10)
160 NEXT
165 GOSUB510
170 B=0:W=0
190 FORZ=1TO4
200 IFC(Z)<>G(Z)THEN230
210 G(Z)=0
220 B=B+1
230 NEXT
240 FORZ=1TO4
250 IFC(Z)=0THEN300
260 FORJ=1TO4
270 IFC(Z)<>G(J)THEN290
280 W=W+1
290 NEXT
300 NEXT
320 PRINT" - ■"B" ■"W
370 IFB=4THEN1000
380 NEXTH
390 GOTO1010
440 END
450 A$=" ■■■■■■■■■■"
460 FORB=3TO8
470 PRINTMID$(A$,B,1);B;"■ ";
475 IFB=5THENPRINT
480 NEXT
490 PRINT"■":PRINT
500 RETURN
510 PRINT"J";
515 FORZ=4TO1STEP-1
520 PRINTMID$(A$,G(Z),1);"■ ";

```



```

530 NEXT
550 RETURN
1000 PRINT"YOU GUESSED IT!"
1010 PRINT"THE CODE WAS:"
1015 PRINT" ";TAB(5);
1020 FORZ=4TO1STEP-1
1030 PRINTMID$(A$,C(Z),1);" ";
1040 NEXT
1050 GETA$:IFA$=""THEN1050
1060 IFA$="S"THENEND
1070 RUN

```

# VICTIM

This game is simplicity itself. You choose your 'victim', a number from one to six, and then sit back and watch the VICTim race get underway.

Lines 15 and 55 are interesting. They allow the VIC to assign a different colour to each number, but the same colour to each number when it is printed. You'll see what I mean when you run it. The sound routine, starting at line 1000, may well benefit from your programming skills.

```

5 POKE36879,8
10 REM VICTIM
12 PRINT" ":PRINT:PRINT:PRINT:PRINT:PRINT
15 A$=" "
20 PRINT " NAME YOUR VICTIM",,,,,," (1 TO 6)";
30 INPUTW
35 IFW<1ORW>6THEN30
40 FORJ=1TO6:B(J)=0:NEXT
50 PRINT" ":REM CLEAR SCREEN
52 PRINT" ":REM HOME
54 FORM=1TO6
55 C$=MID$(A$,M,1)
60 PRINT TAB(B(M));C$ ;M
65 GOSUB1000
70 PRINT
80 B(M)=B(M)+INT(RND(1)+.5)
90 IFB(M)>18THEN120
100 NEXT
110 GOT052
120 FORZ=MT08
130 PRINT
140 NEXT
150 PRINT" END OF RACE "
160 PRINT
170 PRINT" THE WINNER IS"M
180 IFW=MTHENPRINT"AND YOU BACKED IT!":PRINT"WELL DONE"

```

```

185 PRINT
190 IF W<0 THEN PRINT "BUT YOU BACKED" W:PRINT:PRINT "BAD LUCK"
200 PRINT
210 FOR J=1 TO 800: NEXT
220 PRINT "PRESS ANY KEY TO"
230 PRINT "START"
240 PRINT "A NEW RACE"
250 GET V$
260 IF V$=" " THEN 250
270 RUN
1000 POKE 36878, 15
1010 FOR Q=1 TO R(M)
1020 POKE 36876, 180+INT(RND(1)*60)
1030 NEXT Q
1040 POKE 36878, 0
1070 RETURN

```

# MOTORCYCLE JUMP

You are Evil K., world champion stunt rider, and your task in this graphic game on the VIC 20 is to jump over a number of red London buses. Each time you succeed, a new bus is added to the ones you must leap.

When you run the program, you'll be told the angle of the ramp, and the number of buses you must jump. You enter the speed at which you wish to leave the ramp (we suggest you start off between 34 and 40 miles per hour) and you'll see, and hear, your motorbike spin up over the ramp, and – hopefully – over the buses. If you succeed, the ramp will be raised, and the number of buses will be increased.

Because the angle of the ramp affects the distance you can clear, and because this program uses an approximation to genuine formulae to work out if you would clear the buses or not given a particular take-off speed on a ramp of that angle, you'll find your interest in this program will remain high, no matter how many times you play it. There is no way, for example, to know that if there are four buses, a speed of 42 mph will always clear them. It depends on the angle of the ramp.

If you use too much acceleration, you'll flip upside down. Too little, and you'll crash. Far too little, and the program will tell you that you've 'bingled'! If you manage to clear eleven buses, you'll be declared 'The Champ'.

```

10 REM MOTORCYCLE JUMP
12 POKE 36879,24
15 PRINT "Q":U=7
20 GOSUB2000:REM VARIABLES
30 GOSUB3000:REM DISPLAY
100 GOSUB1000:REM CALCULATE
105 A$="BINGLED!"
110 IFABS(F-L)>5 THEN A$="CRASHED"
120 IFF>=LANDF=<L+15 THEN A$="SUCCESS"
130 IFF>L+15 THEN A$="LANDED UPSIDE DOWN"
140 U=1:GOSUB3000
145 L=L+10:N=N+INT(RND(1)*3+1)
147 B=B+1:IFB=11 THEN 5000:REM CHAMPION
200 GOTO100
300 REM SOUND FX
310 POKE36877,220
320 POKE36878,K/3
330 RETURN
360 POKE36877,240
370 POKE36878,1.7*2
380 RETURN
1000 PRINT "SLOPE >"N"DEGREES"
1010 PRINT "A"HERE ARE "B"BUSES"
1020 PRINT "A"THIS IS ATTEMPT "A
1040 INPUT "ASPEED":V:IFV<0 THEN 1040
1045 FORK=1 TO 500: NEXT
1050 A=A+1:E=(V*528)/360
1060 F=(E*COS(N))*((E*SIN(N))/32.2):F=ABS(F)
1070 RETURN
2000 L=10:N=25:B=1:A=1
2030 RETURN
3000 PRINT "SLOPE >"N"DEGREES"
3010 FORZ=1 TO B
3020 PRINT "A";
3030 NEXT:PRINT
3040 IFU=7 THEN RETURN
3050 POKE7944,127:POKE38664,4
3060 FORK=1 TO 50:GOSUB310: NEXT
3070 POKE38664,1
3080 W=7921:W1=38641
3090 FORZ=1 TO B+4
3095 IF4*INT(Z/4)=Z THEN W=W-23:W1=W1-23
3100 POKEW+Z,127
3200 POKEW1+Z,4
3210 FORK=1 TO Z:GOSUB360: NEXT
3220 POKEW1+Z,1
3300 FORK=1 TO 10: NEXT
3310 NEXTZ
3330 IFLEFT$(A$,1)<"S" THEN 4000
3350 FORZ=B+4 TO B+8
3360 W=W+23:W1=W1+23
3370 POKEW+Z,127
3380 POKEW1+Z,4
3390 FORK=1 TO 20+Z
3400 NEXT
3410 POKEW1+Z,1
3420 NEXTZ
3430 POKE36878,0
3450 RETURN
4000 FORK=1 TO 20:POKE36879,INT(16*RND(1)+2+RND(1)*8):GOSUB320: NEXT

```

# MONZA

Monza is a mind-reading mathematical wizard. You first think of a number between one and 63. Monza displays a series of numbers on the screen, and you press 'Y' if your number is among those displayed, 'N' if it is not. After a number of screenfuls of numbers have been shown, with suitably magical sounds, Monza will tell you the number you thought of. This makes a good demonstration program to show the VIC off to your friends.

```

10 REM MONZA, MATHEMATICAL WIZARD
15 B$="■■■"
20 X=.5:N=0
30 POKE36879,25
40 PRINT"YOUTHINK OF A NUMBER"
45 PRINT"BETWEEN 1 AND 63."
50 PRINT"NOI WILL SHOW YOU FIVE"
55 PRINT"SCREENFULLS OF"
60 PRINT"NUMBERS. IF YOUR"
65 PRINT"NUMBER IS ON THE"
70 PRINT"SCREEN, PRESS 'Y' TO"
80 PRINT"TELL ME. IF IT IS NOT"
85 PRINT TAB(8);"PRESS 'N'."
90 PRINT"NOI WILL TELL YOU YOUR"
95 PRINT"NUMBER AT THE END"
100 PRINT"IS YOUR■■■■PRESS ANY KEY..."
110 GET A$:IF A$="" THEN 110

```

```

115 PRINT"END"
120 PRINT"1 3 5 7 9 11 13 15 17"
121 PRINT"19 21 23 25 27 29 31"
122 PRINT"33 35 37 39 41 43 45"
123 PRINT"47 49 51 53 55 57 59"
124 PRINT TAB(8);"61 63"
130 PRINT
140 GOSUB 390
150 PRINT"END"
151 PRINT"2 3 6 7 10 11 14 15"
152 PRINT:PRINT"18 19 22 23 26 27 30"
153 PRINT:PRINT"31 34 35 38 39 42 43"
154 PRINT:PRINT"46 47 50 51 54 55 58"
155 PRINTTAB(8);"59 62 63"
160 PRINT
170 GOSUB 390
180 PRINT"END"
181 PRINT"4 5 6 7 12 13 14 15"
182 PRINT"20 21 22 23 28 29 30"
183 PRINT"31 36 37 38 39 44 45"
184 PRINT"46 47 52 53 54 55 60"
185 PRINTTAB(8);"61 62 63"
190 GOSUB 390
200 PRINT"END"
201 PRINT"8 9 10 11 12 13 14 15"
202 PRINT"24 25 26 27 28 29 30"
203 PRINT"31 40 41 42 43 44 45"
204 PRINT"46 47 56 57 58 59 60"
205 PRINTTAB(8);"61 62 63"
210 GOSUB 390
220 PRINT"END"
221 PRINT"16 17 18 19 20 21 22"
222 PRINT"23 24 25 26 27 28 29"
223 PRINT"30 31 48 49 50 51 52"
224 PRINT"53 54 55 56 57 58 59"
225 PRINTTAB(8);"60 61 62 63"
230 GOSUB 390

```



```

240 PRINT"000"
241 PRINT"32 33 34 35 36 37 38"
242 PRINT"039 40 41 42 43 44 45"
243 PRINT"046 47 48 49 50 51 52"
244 PRINT"053 54 55 56 57 58 59"
245 PRINTTAB(8);"060 61 62 63"
250 GOSUB 390
260 POKE 36878,15
270 FOR T=1TO255
280 POKE 36876,T
290 POKE 36876,255-T/2
300 NEXT
310 POKE 36878,0
320 POKE 36876,0
330 PRINT"00 YOUR NUMBER WAS"
340 FOR T=1TO4000:NEXT:RUN
390 X=X+X
395 C$=MID$(B$(INT(RND(1)*3)+1),1)
400 POKE36878,15
410 FOR T=100 TO 255
420 POKE 36876,T-N*3
430 NEXT T
440 POKE 36878,0
450 POKE 36876,0
455 PRINT
460 PRINT C$"IF YOUR NUMBER IS"
470 PRINT"HERE PRESS 'Y', IF"
480 PRINT "NOT, PRESS 'N'."
490 GET A$
500 IF A$="" THEN 490
510 IF A$<>"N" THEN N=N+X
520 FOR T=1 TO 500
530 NEXT T
540 RETURN

```

# MENTO

Mento is another mathematical mind-reading wizard. Just get your friends to follow the instructions to be amazed, pressing any key after you have followed the wizard's wise words.

```
10 REM MENTO
15 POKE 36879,8:PRINT"#"
20 GOSUB1020
30 PRINT"MULTIPLY YOUR AGE BY"
40 PRINT"TWO, THEN ADD FIVE"
50 GOSUB1000
60 PRINT"NOW MULTIPLY THAT"
70 PRINT"BY FIFTY, AND"
80 PRINT"SUBTRACT 365"
90 GOSUB1000
100 PRINT"NOW ADD THE AMOUNT"
110 PRINT"OF CHANGE IN YOUR"
120 PRINT"POCKET"
130 GOSUB1000
140 PRINT"NOW GIVE ME THE"
150 PRINT"NUMBER YOU'VE ENDED"
160 PRINT"UP WITH"
170 INPUTA:A=A+115
180 BX=A/100
190 A=A-BX*100
200 PRINT"YOU HAVE"A"CHANGE."
210 PRINT
220 PRINT"YOU ARE"BX"YEARS OLD"
230 END
1000 GETA$
1010 IFA$=""THEN1000
1020 PRINT"J"
1030 PRINT"XXX"
1040 RETURN
```

# TIME GUARDIAN

In this challenging game written by Adam Burbidge of Crawley you have to recover the six parts of the key of time. The ship starts with 700 energy points, and you lose five points for each move.

When you first run it, you'll see the message I AM CREATING A GALAXY while the VIC sets up the universe for your game. After a brief delay, the screen will clear, and it is Decision Time. Every so often the VIC will ask you to set co-ordinates, and then it will print:

QUADRANT?	You enter a number from 0 to 6
SECTOR?	You enter another number from 0 to 6
This is repeated.	Then it will print:
TIME?	You enter either 0 or 1

If you land on a hostile planet, the VIC will print SHIP UNDER ATTACK, then SHIP ENERGY. . . and a number. You need to press any key (which you may have to do several times rapidly) to escape from the hostile planet.

From time to time you'll find part of the 'key of Chronos', the point of the whole game. You must find all six parts of it to win, and you must find them before your ship's energy is exhausted.

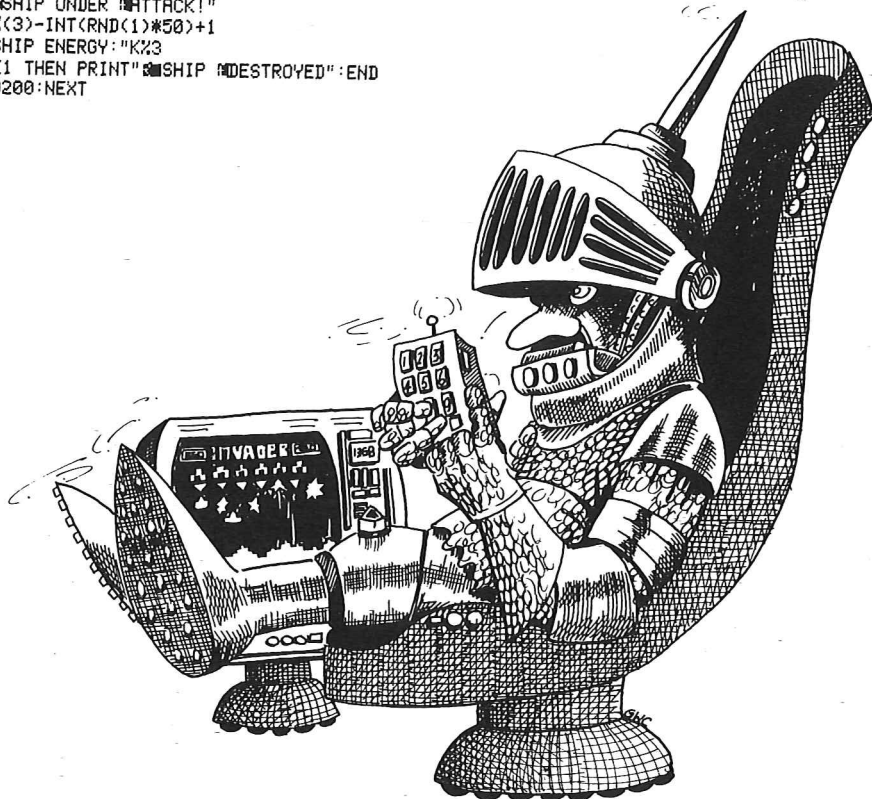
```
3 REM TIME GUARDIAN
4 REM BY ADAM BURBIDGE
5 DIM K$(4), Z$(1), A$(6,6,1)
6 K$(3)=700:F=75
10 PRINT"J":REM CLEAR SCREEN
20 PRINT"  I AM CREATING A GALAXY"
30 B=INT(RND(1)*6)
40 C=INT(RND(1)*6)
50 D=INT(RND(1)*6)
60 E=INT(RND(1)*2)
70 A$(B,C,D,E)=2
75 F=F-1:IF F<0 THEN 85
80 GOTO30
85 PRINT"J":REM CLEAR SCREEN
90 GN=INT(RND(1)*2)+1
100 PRINT"  YOU START AT GLODNOR"
105 PRINT
110 PRINT"  QUADRANT 5"
115 PRINT
120 PRINT"  SECTOR 3,4"
```



```

125 PRINT
130 PRINT"TIME"GN*1000"EARTH YEARS"
135 PRINT
140 PRINT"SET CO-ORDINATES"
150 INPUT"QUADRANT";G
160 INPUT"SECTOR";H
165 PRINT" "
170 INPUT"SECTOR";I
180 INPUT"TIME";J
185 K%(3)=K%(3)-5
190 IFA%(G,H,I,J)=1 THEN PRINT"YOU ARE ON GLODNOB":GOTO90
200 IFA%(G,H,I,J)>1 THEN 300
210 PRINT"DEEP SPACE":GOTO140
300 PRINT"YOU ARE ON A PLANET"
310 PRINT"QUADRANT"G
320 PRINT"SECTOR"H,I
330 PRINT"TIME"J*INT(RND(1)*7)
340 K%(1)=INT (RND(1)*20)+1
350 IFK%(1)>5 THEN 370
360 PRINT"UNINHABITED":GOTO140
370 IFK%(1)>10 THEN 400
380 PRINT"INHABITANTS HOSTILE":GOTO500
400 PRINT"PART OF THE KEY OF"," CHRONOS "
410 K%(2)=K%(2)+1
420 PRINT"YOU NOW HAVE"K%(2)"PARTS","OF THE KEY"
430 IFK%(2)=6 THEN PRINT"YOU HAVE WON!":END
440 GOTO140
500 GET Z%(1)
510 IF Z%(1)=""THEN540
520 GOTO140
540 PRINT"SHIP UNDER ATTACK!"
550 K%(3)=K%(3)-INT(RND(1)*50)+1
560 PRINT"SHIP ENERGY:"K%3
565 IFK%(3)<1 THEN PRINT"SHIP DESTROYED":END
570 FORT=1TO200:NEXT
580 GOTO500

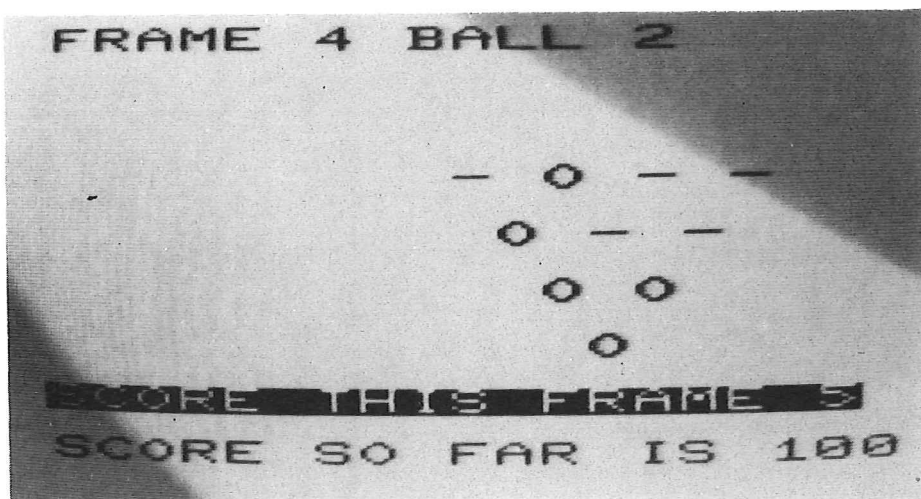
```



# SUPERBOWL

The VIC sets up the pins for you in this simulation of a 10-pin bowling alley, and then obligingly knocks them down, adding up your score as it goes.

There are 10 frames to a game, two balls to a frame. As you'll play this game, you'll find that strikes are few and far between. There is a highest score routine (310 to 330) which you may well want to adapt for other games you write.



```

5 REM SUPERBOWL
20 Y=0:S=0
30 PRINT"J":REM CLEAR
40 FORB=1TO10
50 GOSUB500
60 FORC=1TO2:PRINT:PRINT:PRINT
65 PRINT"J":REM HOME
70 PRINT"  FRAME"B"  BALL"E
75 PRINT:FORC=1TO500:NEXT
80 Z=0
90 FORC=1TO10
100 IFE=2THEN120
110 A(C)=79
120 IF RND(1)>0.5 THEN A(C)=45
130 IF A(C)=45 THEN Z=Z+1
140 NEXTC
145 PRINT:PRINT:PRINT
150 PRINT,"■"CHR$(A(10))" "CHR$(A(9))" "CHR$(A(8))" "CHR$(A(7))
155 PRINT:GOSUB700
160 PRINT," "CHR$(A(6))" "CHR$(A(5))" "CHR$(A(4))
165 PRINT:GOSUB700
170 PRINT," "CHR$(A(3))" "CHR$(A(2))
175 PRINT:GOSUB700

```

```

180 PRINT, "  CHR$(A(1))
190 PRINT
200 PRINT "  SCORE THIS FRAME"Z"
215 PRINT
220 IFE=1 ANDZ=10 THEN 370
230 IFZ=10 THEN Z=15:PRINT, "  BONUS!!"
240 IFE=2 THEN S=S+Z
250 IFE=2 AND B<10 THEN PRINT "  SCORE SO FAR IS"S
260 FORT=1TO1000:NEXT
270 PRINT
280 NEXTE
290 NEXTB
300 PRINT"SCORE FOR THAT GAME      WAS"S"
310 IFS<Y THEN 330
320 Y=S
330 PRINT "  HIGHEST SO FAR"Y
340 FORT=1TO9000:NEXT
350 S=0
360 GOTO30
370 PRINT
380 PRINT, "  STRIKE!!"
385 PRINT
390 FORT=1TO1000:NEXT
410 S=S+15
420 E=2
440 GOTO230
500 FORD=1TO10:A(D)=79:NEXT
505 GOSUB2000
510 PRINT "8"
520 PRINT "  FRAME      "
530 PRINT:PRINT
540 PRINT:PRINT
550 PRINT, "  "CHR$(A(10))" "CHR$(A(9))" "CHR$(A(8))" "CHR$(A(7))
555 PRINT
560 PRINT, "  "CHR$(A(6))" "CHR$(A(5))" "CHR$(A(4))
565 PRINT
570 PRINT, "  "CHR$(A(3))" "CHR$(A(2))
575 PRINT
580 PRINT, "  "CHR$(A(1))
585 FORJ=1TO5:PRINT "      ":NEXT
590 FORT=1TO1000:NEXT
600 RETURN
700 REM DELAY, SOUND
710 POKE36878,15
720 FORU=249TO239STEP-2
730 POKE36876,U
750 FORU=238TO249
760 POKE36876,U
770 NEXT
780 POKE36876,0
800 POKE36878,0
810 FORT=1TO50:NEXT
820 RETURN
2000 REM SET UP SOUND
2010 POKE36878,10
2020 FORK=248TO148 STEP-2
2025 POKE36879,26+INT(RND(1)*6)
2030 POKE36876,K
2040 POKE36875,K
2050 NEXT
2055 POKE36878,0
2060 RETURN

```

# GOMOKU

In this game, you and the VIC take it in turns to place your pieces ('H' for you, 'C' for the computer) on an eight by eight board, trying to get five pieces in a row. The computer plays an extremely good game, so good that it was necessary to cheat when we were developing this program in order to win to check that the YOU WIN routine was working correctly.

You have the choice of first or second move, and you enter your move by indicating the number along the side, then across the top.

As you can see, the program cycles through a loop (lines 60 to 100) over and over again, calling up various subroutines to do the needed tasks. The board is printed by lines 230 to 320, and you may well want to modify this to print different symbols for the pieces, or use the same symbols as in the program, but with different colours.

The game can end in three ways. You defeat the computer (line 640), the computer defeats you (line 1140) or the computer concedes defeat (which it will only do after having chosen 400 squares at random, see lines 950 to 975). This program is based on one written by Graham Charlton.

```
10 REM GOMOKU
15 REM BASED ON PROGRAM BY GRAHAM CHARLTON
20 GOSUB1180
30 GOSUB230
60 GOSUB420
70 GOSUB230
80 GOSUB540
90 GOSUB230
100 GOSUB1030
110 GOTO60
140 E=A
145 E=E+N:IFA(E)>2THENRETURN
150 K=K+1:GOTO145
230 PRINT"XXXXXXXXXX 12345678"
240 FORA=1TO8:PRINT"XXXXXXXXX";A;
270 FORB=2TO9:N=A(A*10+B)
280 IFM=CTHENPRINT"XC";
290 IFM=HTHENPRINT"XH";
300 IFM=46THENPRINT". ";
310 NEXT:PRINT"XXXXXXXXX";
320 PRINT"XXXXXXXXXX 12345678XXXX"
360 RETURN
420 INPUTG:G=G+1:IFG<12ORG>89ORA(G)>46THEN420
480 Z=H:A(G)=Z:RETURN
540 A=G:L=0:FORX=1TO4:K=0:N=X(X)
590 GOSUB140
600 N=-N:GOSUB140
610 IFK>LTHENL=K
630 NEXT
640 IFL>3THENPRINT"XXXXXXXXXXXXXXXXXXXXYOU WIN":GOTO640
650 T=1
660 IFT>2THENZ=C
670 IFT=2THENZ=H
```

```

680 G=0:H1=0:L=0
710 FORA=12TO89:M=0:IFA(A)<>46THEN900
740 FORX=1TO4:K=0:N=X(X):GOSUB140
780 N=-N:GOSUB140
800 IFK>LTHENH1=0:L=K
810 IFL<>KTHEN860
820 IFT=1ANDL<4OR(T=2ORT=3)ANDL<2THEN860
850 M=M+1
860 NEXT
870 IFM<=H1THEN900
880 H1=M:G=A
900 NEXT
910 IFH1<>0THEN980
920 T=T+1:IFT<>4THEN660
940 A=1
950 G=INT(RND(0)*77)+13
960 IFA(G)=46THEN980
970 A=A+1:IFAC400THEN950
975 PRINT"XXXXXXXXXXXXXXXXXXXX CONCEDE":GOTO975
980 A(G)=C:RETURN
1030 Z=C:A=G:L=0
1060 FORX=1TO4:K=0:N=X(X)
1090 GOSUB140
1100 N=-N:GOSUB140
1120 IFK>LTHENL=K
1130 NEXT
1140 IFL>3THENPRINT"XXXXXXXXXXXXXXXXXXXX WIN":GOTO1140
1150 RETURN
1180 PRINT"J"
1200 DIMA(100)
1210 FORA=1TO8:FORB=2TO9:A(A*10+B)=46:NEXTB,A
1260 FORQ=1TO4:READX(Q):NEXT
1290 DATA1,9,10,11
1300 H=ASC("H"):C=ASC("C")
1340 PRINT"XXXXENTER 'Y' IF YOU WANT THE FIRST MOVE, 'N' IF YOU DON'T"
1350 GETF$:IFF$<>"N"ANDF$<>"Y"THEN1350
1380 PRINT"J":POKE36879,255
1390 IFF$="Y"THENRETURN
1400 FORJ=1TOINT(RND(0)*12)+1
1410 READZ:NEXT
1440 A(Z)=C:RETURN
1450 DATA34,35,36,44,46,47,54,55,56,57,66

```



# FAIRWAY

In this game, you are on a nine hole golf course. The length of each hole varies each time you play the game. The program makes much use of POKEing to create the green, and move the ball. The sound the ball makes is particularly interesting if you overshoot the hole.



```
10 REM FAIRWAY
15 POKE36879,28
20 SC=0:REM SCORE
30 AV=0:REM AVERAGE
100 FORA=1TO9:REM NO. OF HOLES
105 T=0:REMTALLY THIS HOLE
110 D=INT(RND(1)*7)+14:REM DISTANCE TO HOLE
120 PRINT"T":REM CLEAR SCREEN
125 C=0
130 M=0
400 GOSUB6000
500 GOSUB5000
510 IFM<>D THEN 500
520 IFABS(M-D)<2 THEN 8000
525 IFM>21 M=21
530 GOTO500
540 END
5000 PRINT "HOLE NUMBER"A
5005 PRINT,D*10"YARDS"
5010 PRINT
5012 PRINT"SCORE"SC"TALLY"T
5015 IFD=M THEN 8000
5020 PRINT"DISTANCE FROM HOLE:"
5025 IF D-M>10 THEN PRINT"(D-M)*10"YARDS "
5026 IF D-M< 10 THEN PRINT " (D-M)*10"YARDS "
5030 PRINT"
```

```

5040 INPUT"WHICH CLUB";C
5042 T=T+1
5044 GOSUB7000
5045 POKE38730+M,1
5047 IFM>D THENC=-C
5060 M= INT(M+C/3*RNDRND(1+1))
5065 C=0
5070 POKE8010+M,81
5080 POKE38730+M,2
5090 RETURN
5999 END
6000 REM DRAW GREEN
6010 FORQ=3032TO8119
6020 POKEQ,102
6030 POKEQ+38752-8032,5
6040 NEXT Q
6050 POKE8032+D,74:POKE8033+D,75
6060 POKE38752+D,2:POKE38753+D,2
6065 POKE8010+M,81
6070 POKE38730+M,2
6100 RETURN
7000 FORW=128TO190+3*M
7005 POKE36878,15
7020 POKE36875,W
7030 POKE36874,W
7040 NEXTW
7050 POKE36875,0
7060 POKE36874,0
7500 RETURN
8000 T=T-1
8002 POKE36876,220
8003 FORP=1TO5:NEXT
8004 POKE36876,0:POKE36876,225:POKE36876,0
8010 PRINT"YOU GOT IT IN" T
8020 POKE8010+M,1
8030 POKE8032+M,81
8040 POKE38774+M,2
8050 POKE8033+M,102
8060 POKE38775+M,5
8062 SC=SC+T
8065 PRINT"■AVERAGE"INT(SC/A+.5)
8070 FORH=1TO5000:NEXT
8990 NEXTA
9000 END

```



# ROBOT NIM

The VIC prints a number of odd triangle shapes, tells you how many you can take, then takes a few of the shapes itself. This fascinating process continues until there is only one left. At this point the player who must take the last one loses.

## Structure of the program

Line 20: Determines the number of objects.

Line 50: Determines maximum number to be taken.

Lines 100 to 130: Print out the correct number of triangles, starting new lines at random (see line 120).

Lines 210 and 215: Determine how many the 'robot' will take. The random bit at the end of line 210 ensures that the robot makes mistakes from time to time. Line 215 ensures that the mistake is not so large that the robot breaks the rules.

Lines 280 to 290: Place a coloured border between rounds of the game. Design this to suit yourself.

```
10 REM ROBOT NIM
20 M=0:E=0:Z=16+INT(RND(1)*8)
30 PRINT" ":REM CLEAR SCREEN
40 IF 2*INT(Z/2)=Z THEN Z=Z+1
50 H=3+INT(RND(1)*2)
60 PRINT"  MAXIMUM TO TAKE" H
70 GOSUB280
80 IF E>0 THEN PRINT"  YOU TOOK"E,"  ROBOT TOOK"Q
90 PRINT"  "
100 PRINT"  ":FOR K=1 TO Z
110 PRINTK;" ";
120 IF RND(1)>0.6 THEN PRINT
130 NEXT K
140 GOSUB280
145 PRINT"  HOW MANY WILL","YOU TAKE";
150 INPUT E
160 IF E>H THEN 150
170 Z=Z-E
180 GOSUB280
190 IF Z<1 THEN PRINT"  YOU TOOK THE LAST ONE","  SO ROBOT WINS!!  ":END
210 Q=Z-1-INT((Z-1)/(H+1))* (H+1)-INT(RND(1)*2)+INT(RND(1)*2)
215 IF Q<1 OR Q>H THEN 210
220 GOSUB280
230 Z=Z-Q
240 IF Z=0 THEN PRINT"  ROBOT TOOK"Q,,,"  SO YOU  WIN!!":END
270 GOTO60
280 PRINT
285 PRINT"#####
290 RETURN
```

# ZAUPER ATTACK

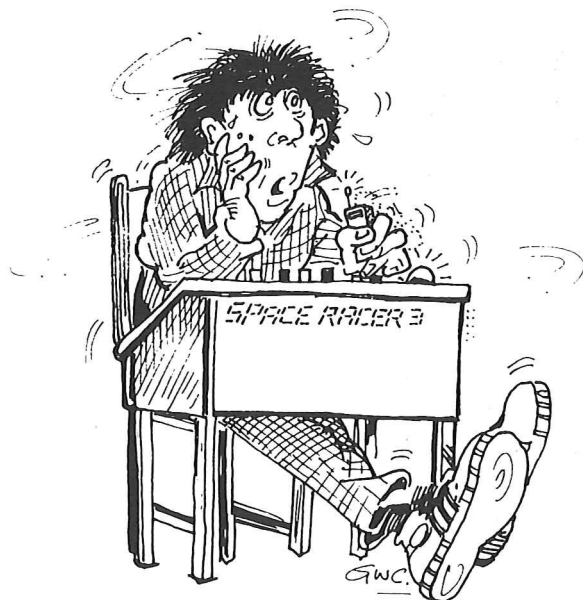
In this program, by Christopher Hutber of Milton Keynes you must zap the Klingons before they are eaten by the Zaupers. you need to take care because these creatures may also eat you. "Z" moves your gun left, and "C" moves it right. "X" will fire your gun. Full instructions are within the program.

```
0 PRINT"!"
1 REMARKABLE PROGRAMME BY CHRISTOPHER HUTBER AGED 12
2 REM 24-27/12/81
3 GOSUB 7000
4 T1=T1
5 SC=0
6 PRINT"!"
7 PRINT"XXXXXXXXXXXXXXXXXXXX"
8 REM DRAW KLINGONS
9 POKE 36879,8
10 FOR I=38400 TO 38531
11 POKE I,1
12 NEXT I
13 FOR H=7680 TO 7767STEP 3
14 POKE H,81
15 NEXT H
16 REM DRAW GUN
17 T=7955
18 FOR R=38664TO38664+21
19 POKE R,7
20 NEXT R
21 POKE T,65
22 IF SC=30 THEN 1000
23 REM GET INSTRUCTION
24 GET A$:IF A$=""THEN 3000
25 IF A$="X"THEN400
26 IF A$="C"THEN300
27 IF A$="Z"THEN200
28 IF A$="*"THEN 5000
29 GOTO 140
30 REM MOVE GUN
31 FOR B=7944TO7965
32 POKE B,70
33 NEXT B
34 T=T-1:GOTO 120
35 FOR B=7944TO7965
36 POKEB,70
37 NEXTB
38 T=T+1:GOTO 120
39 REM FIRE
40 FORF=TT07680+(T-7944)STEP-22
41 POKEF,46
42 FOR W=1 TO25:NEXTW
43 POKE F,32
44 IF PEEK(F-22)=81THEN GOTO 480
45 NEXTF
```

```

455 POKE F+22,96
460 FOR I=1TO50:NEXT I
470 POKE F,87
472 FOR G=7680+(T-7944)TOT-22STEP22
474 POKE G,96
480 POKE F-22,32
482 IF PEEK(F)<81AND F<7702 THEN GOTO 600
488 POKE 36877,220
490 L=7
500 POKE 36878,L
510 FOR M=1TO 200
520 NEXT M
540 POKE 36876,0
550 POKE 36878,0
555 SC=8C+1
560 PRINT"SCORE:";SC;"T:";GOTO 120
600 PRINT"SCORE:";SC;"T:";GOTO 120
1000 PRINT"T"
1005 T2=T1
1010 POKE 36879,30
1020 PRINT"WELL DONE"
1030 PRINT"YOU WIPED OFF THE      KLINGONS IN ";INT((T2-T1)/60);" SECONDS!"
1031 POKE 36878,15
1032 FOR L=1 TO 100
1033 POKE 36876,INT(RND(1)*128)+128
1034 FOR M=1 TO 10
1035 NEXT M
1036 NEXT L
1037 POKE 36876,0
1038 POKE 36878,0
1040 PRINT:PRINT
1050 PRINT"PLAY AGAIN ?"
1055 PRINT:PRINT
1060 GET S$
1070 IF S$="" THEN 1060
1080 IF S$="Y" THEN 8
1090 IF S$="N" THEN 1200
1100 GOTO 1060
1200 PRINT TAB(7);"BYE BYE"
1205 PRINT
1210 PRINT"THANKS FOR PLAYING"
1300 END
2999 REM ACTIVATE ZAUPERS
3000 FORE=1TO10:R=INT(RND(1)*75)+7680
3010 IF R>7767 THEN 3000
3020 Q=INT(RND(1)*20)+7944
3050 D=0.5+RND(1)
3060 IF D<1 THEN V=21
3070 IF D>1 THEN V=23
3100 FOR Z=RTOQSTEP V
3110 POKE Z,61
3120 POKE Z-V,32
3130 FOR X=1TO25:NEXT X
3150 IF A$="" THEN 5150
3200 GET A$:IF A$<>" " THEN 170
3300 NEXT Z
3400 IF Z=TT THEN 4000
3500 GOTO 170
4000 PRINT"ZAUPERS HAVE WON "
4050 PRINT
4100 PRINT"MOON BASE DESTROYED!"
4150 REM EXPLOSION
4200 POKE 36877,220

```



```

4230 FOR L=15T00STEP-1
4250 POKE 36879,L
4270 FOR M=1T0300
4290 NEXT M
4300 NEXT L
4320 POKE 36877,0
4330 POKE 36878,0
4500 GOTO 1040
5000 SC=SC-5
5100 PRINTSC;"J"
5200 FOR K=1T0 50:NEXT K
5300 GOTO 10
7000 PRINTTAB(3);"  "
7010 PRINT TAB(3);"I ZAUPER ATTACK! "
7020 PRINTTAB(3);"  BY  "
7030 PRINTTAB(3);"I C.R.J. HUTBER  "
7040 PRINTTAB(3);"  "
7060 PRINT:PRINT
7100 PRINT"INSTRUCTIONS (Y OR N)?"
7120 GET M$:IF M$=""THEN 7120
7130 IF M$="N"THEN RETURN
7140 IF M$="Y"THEN 7500
7150 GOTO 7120
7500 PRINT"YES"
7510 PRINT
7520 PRINT"'KLINGON ATTACK' IS A GAME OF SPEED."
7530 PRINT:PRINT"THE OBJECT IS TO KILL ALL THE KLINGONS IN  AS QUICK TIME AS
POSSIBLE."
7540 PRINT:PRINT
7550 PRINT"HIT A KEY"
7560 GET N$:IF N$=""THEN 7560
7600 PRINT"J"
7610 POKE 36879,0
7620 PRINT"  KEY"
7630 PRINT"  "
7640 PRINT:PRINT
7650 PRINT"  -KLINGON"
7655 PRINT
7660 PRINT"  -ZAUPER"
7665 PRINT
7670 PRINT"  -YOUR GUN"
7672 PRINT:PRINT:PRINT
7675 PRINT"HIT A KEY"
7680 GET O$:IF O$=""THEN 7680
7700 PRINT"J"
7710 POKE 36879,234
7720 PRINT"THE ZAUPERS EAT THE  KLINGONS AND WILL"
7730 PRINT"CRASH INTO YOU GIVEN  HALF A CHANCE."
7735 GOSUB 9000
7740 PRINT:PRINT
7750 PRINT"TO MOVE YOUR GUN PRESS"
7760 PRINT:PRINT"'Z'-MOVES IT LEFT"
7770 PRINT:PRINT"'C'-MOVES IT RIGHT"
7800 PRINT:PRINT
7810 PRINT"TO FIRE GUN PRESS:'X'"
7820 PRINT:PRINT
7830 PRINT"HIT A KEY"
7840 GET V$:IF V$=""THEN 7840
7850 PRINT"J"
7900 PRINT"WHEN ALL THE KLINGONS ON THE SCREEN HAVE"
7910 PRINT"GONE,PRESS:'*'"
7920 PRINT"THIS WILL TAKE YOU  BACK TO THE START AND"
7930 PRINT"TAKE AWAY FIVE POINTS"

```

```

7950 PRINT:PRINT
8000 PRINT"HIT A KEY TO START"
8010 GET X$:IF X$=""THEN 8010
8020 RETURN
9000 PRINT:PRINT
9010 PRINT"THE ZAUPERS ARE HIGHLYSENSITIVE TO THE HEAT "
9020 PRINT"GIVEN OFF BY YOUR      MOVEMENT AND FIRING,"
9030 PRINT"THUS WHEN YOU DO SO,"
9040 PRINT"THEY ARE PARALYSED!"
9045 PRINT:PRINT"HIT A KEY"
9050 GET Y$:IF Y$=""THEN 9050
9060 PRINT"J"
9070 RETURN

```

# VIC-ET-UN

In this program you and the VIC (with thunderstorms, lighting bolts and other distractons) take it in turns to roll a dice trying to get a total as close as possible to – or equal to – 21, without going over 21. This game is a dice version of Blackjack.

It is very simple to play. After pressing RUN, you'll be told to touch the "Z" key to roll the dice, the "M" key to stand, that is, to stay with the total you now have. Your total will apear, and the choice to roll or stand again. Once you've decided to stand, the VIC will start to roll, and will – in two games out of three, on average – beat you.

## Structure of the program

Line 10: Sets the counters for the human (H) and computer (C) scores.  
 Line 20: Sends action to the wild and crazy subroutine which selects colours and sounds randomly.  
 Lines 30 to 80: Accepts and acts on the player's decisions.  
 Lines 90 to 130: Computer decides whether to add to its total or stand.  
 Lines 140 to 250: Decides who has won, creates some more mayhem, starts a new game.  
 Line 260: Delay loop, called various times during the program.  
 Line 990: Assigns A\$.  
 Lines from 1000: Selects an element from A\$ to change the colour, then does some quite exciting things with the colour and sound.

```

5 REM VIC-ET-UN
7 PRINT"J":REM CLEAR SCREEN
10 H=0:C=0
20 GOSUB990
30 PRINTB$;"  ENTER Z TO ROLL":GOSUB1000:PRINTB$;"      M TO STAND"
35 GETM$:IFM$=""THEN35
40 IFM$="M"THEN90
50 H=H+INT(RND(1)*6)+1
60 GOSUB260:GOSUB1000
70 PRINT:PRINTB$;"  YOUR TOTAL IS"H
75 GOSUB260:GOSUB260:PRINT
80 GOTO30
90 IFC>H AND C<22 OR C>21 OR H>21 ORH=21 ANDC=21THEN 140
100 C=C+INT(RND(1)*6)+1
110 GOSUB260

```

```

115 PRINT:PRINT
120 GOSUB1000:PRINTB$;" MY TOTAL IS"C
130 GOTO90
140 GOSUB1000:PRINTB$;
150 GOSUB260:GOSUB1000:GOSUB260
170 IFH=COR<H>21ANDC<21>THEN240
175 PRINT:PRINT:PRINTB$;
180 IF<C>H ORH<21>ANDC<22> THENPRINT,"I";
190 IF<C<HORC>21>ANDH<22>THENPRINT,"YOU";
200 GOSUB1000:PRINTB$;" WIN!!"
210 GOSUB260
220 GOSUB1000
225 FORJ=1TO7:GOSUB260:NEXT
226 GOSUB1000
230 PRINT"J":RUN
240 GOSUB1000:PRINTB$;"DEAD HEAT, BUDDY!"
250 GOTO225
260 FORE=1TO500:NEXT
270 RETURN
990 A$="███"
1000 B$=MID$(A$,INT(RND(1)*7)+1,1)
1010 FORJ=1TOINT(RND(1)*10)+1
1020 POKE36878,INT(RND(1)*10)+6
1030 FORK=1TOINT(RND(1)*7)+2
1040 POKE36876,K+180-INT(RND(1)*30)
1050 NEXTK,J
1070 POKE36879,INT(RND(1)*3)+28:POKE36879,8
1075 POKE36879,INT(RND(1)*3)+28
1080 POKE36879,INT(RND(1)*3)+28
1090 POKE36878,0
1100 RETURN

```

# EVOLUTION

In 1970, John Conway, who was then attending Cambridge University, invented the game of LIFE, which rapidly became a computer 'hit' after it was described in the October 1970 issue of *Scientific American*.

LIFE, which simulates the birth, growth and death of a cell colony, produces rather splendid visual effects.

Mr Conway's rules are simple, but produce unpredictably interesting results. The cells in LIFE are born, live or die in accord with the dictates laid down by the God Conway. According to his writ, each cell has eight neighbours; every cell with two or three neighbours survives to the next generation; if there are three, and only three, neighbouring cells, a new cell is born; and any cell with four or more neighbours dies from overpopulation.

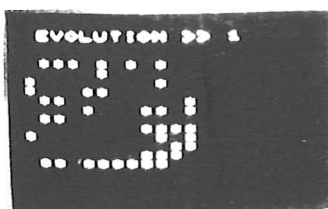
EVOLUTION is a version of LIFE written especially for the VIC. Setting up a black



screen with a red border, the program puts cells on a 14 x 14 grid in accord with the Conway rules. The game as listed is silent, but if you'd like to add a little sound—which tells you how things are going in the program — add the following lines:

```
6   POKE 36878,7
225 POKE 36876,240:POKE 36876,0
1075 POKE 36876,235:POKE 36876,0
```

```
5 REM EVOLUTION
7 PRINT"J":REM CLEAR          SCREEN
8 PRINT"33"
9 POKE 36879,10:REM RED BORDER, BLACK SCREEN
10 DIMA(14,14),B(14,14)
20 FORX=2 TO 13
30 FORY=2 TO 13
50 IF RND(1)>0.5 THEN A(X,Y)=1
60 B(X,Y)=A(X,Y)
70 NEXTY
80 NEXTX
90 GOSUB1000
95 G=G+1
100 FORX=2TO13
110 FORY=2TO13
120 C=0
130 IF A(X-1,Y-1)=1 THEN C=C+1
140 IF A(X-1,Y)=1 THEN C=C+1
150 IF A(X-1,Y+1)=1 THEN C=C+1
160 IF A(X,Y-1)=1 THEN C=C+1
170 IF A(X,Y+1)=1 THEN C=C+1
180 IF A(X+1,Y-1)=1 THEN C=C+1
190 IF A(X+1,Y)=1 THEN C=C+1
200 IF A(X+1,Y+1)=1 THEN C=C+1
210 IF A(X,Y)=1 AND C=2 AND C<3 THEN B(X,Y)=0
220 IF A(X,Y)=0 AND C=3 THEN B(X,Y)=1
230 NEXTY
240 NEXTX
250 GOTO90
999 END
1000 PRINT"J":REM HOME
1003 PRINT:PRINT"  EVOLUTION >>"G;"M":REM WHITE, YELLOW
1005 PRINT
1010 FORX=1TO14
1020 FORY=1TO14
1030 A(X,Y)=B(X,Y)
1040 IF A(X,Y)=1 THEN PRINT"*";
1050 IF A(X,Y)=0 THEN PRINT" ";
1060 NEXTY
1070 PRINT
1080 NEXTX
1090 RETURN
```



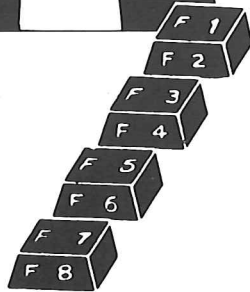
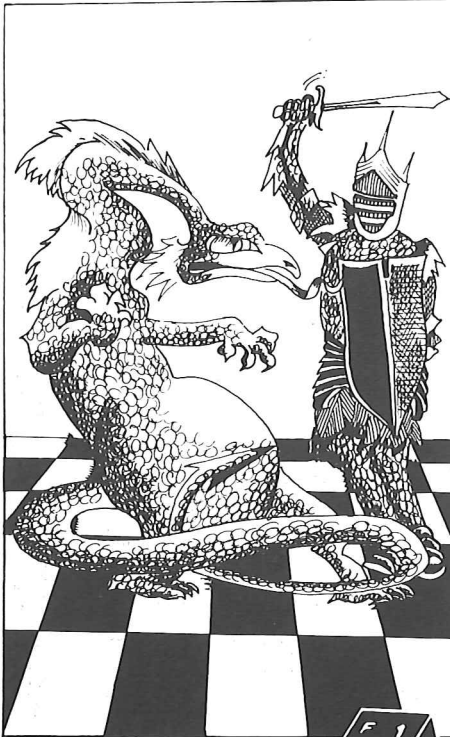
# SUPERPOET

The VIC turns into Wordsworth, cranking out some splendid poems, and a few which are not so brilliant. Once you've seen it in action, change the words in the subroutines 100 to 270 to suit your own taste. Note the use of ON. . .GOSUB in line 40.

```

5 REM SUPERPOET
10 PRINT " ": REM CLEAR SCREEN
15 IF RND(1)>.3 THEN 30
20 FOR J=1 TO RND(1)*2:PRINT:NEXT
30 FOR J=1 TO RND(1)*5+1:PRINT " ":NEXT
35 J=INT(RND(1)*10)+1
40 ON J GOSUB 100,110,120,130,140,150,160,170,180,190,200,210,220,230,240,250,260,270

```



```

50 FOR P=1 TO RND(1)*2000:NEXT
70 GOTO 15
100 PRINT "DETACHED":RETURN
110 PRINT "UNAWARE":RETURN
120 PRINT "UNABLE":RETURN
130 PRINT "INITIATE":RETURN
140 PRINT "REACHED OUT FOR":RETURN
150 PRINT "AVOIDS":RETURN
160 PRINT "SUFFERS":RETURN
170 PRINT "CAPACITY":RETURN
180 PRINT "SPIRITUAL":RETURN
190 PRINT "ALTHOUGH":RETURN
200 PRINT "DISCIPLE":RETURN
210 PRINT "HE":RETURN
220 PRINT "AND":RETURN
230 PRINT "THEN":RETURN
240 PRINT "MASTER":RETURN
250 PRINT "AFTER":RETURN
260 PRINT "LONELY":RETURN
270 PRINT "HIS":RETURN

```

AND THEN  
UNAWARE  
UNABLE

ALTHOUGH  
AND MASTER  
REACHED OUT FOR  
UNABLE

MASTER RETURNED  
UNABLE  
CAPACITY SPIRITUAL  
DISCIPLE HE  
LONELY HE

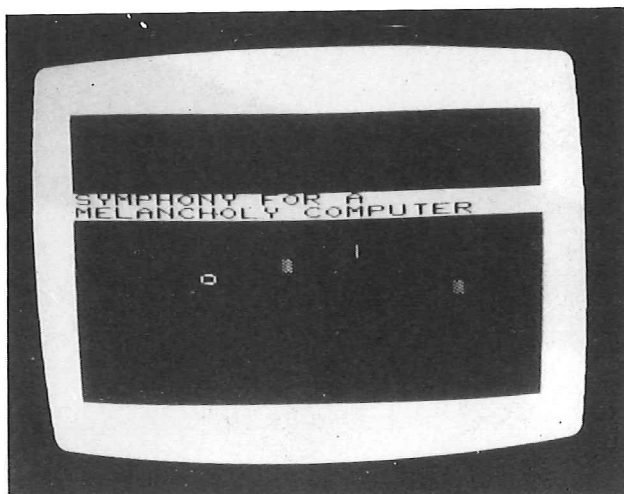
AVOIDS LONELY MASTER

# SYMPHONY FOR A MELANCHOLY COMPUTER

This, the title piece of our book, reflects an effort to write a program which would 'write music' that had some underlying form. The title is obvious and appropriate, once you've heard the computer's output from this program.

The main structure of the music comes from the variables A, B, C, D, E and F which are defined on each run. The program is designed so that the chords change with reference to a measure (E) and so that the changes overlap each other, thus getting rid of harsh stops and starts.

```
10 REM SYMPHONY FOR
15 REM A MELANCHOLY
16 REM COMPUTER
18 GOSUB1000
20 A=128+INT(RND(1)*128)
30 B=128+INT(RND(1)*128)
40 C=128+INT(RND(1)*128)
50 D=128+INT(RND(1)*128)
60 E=INT(RND(1)*4)+1
70 F=2*E
75 POKE36878,F-1
90 FORH=1TO5*E
100 POKE36874,A
110 NEXTH
120 FORH=1TO5*E
130 POKE36875,B
140 NEXTH
150 FORH=1TO5*E
160 POKE36876,C
170 NEXTH
180 FORH=1TO5*E
190 POKE36877,D
200 NEXTH
```



```

210 FORH=1T02*E
220 POKE36874,A
230 NEXTH
240 FORH=1T05*E
250 POKE36876,C
260 NEXTH
500 RUN
1000 PRINT"J":REM CLEAR
1010 POKE36879,16*(INT(RND(1)*16))+(INT(RND(1)*8))+8
1020 RETURN

```

## Symphony for a Melancholy Computer with Zen Show

This is the same program as just listed, with an underplayed visual display added.

```

10 REM SYMPHONY FOR
15 REM A MELANCHOLY
16 REM COMPUTER
17 REM WITH ZEN SHOW
18 GOSUB1000

```

```

20 A=128+INT(RND(1)*128)
30 B=128+INT(RND(1)*128)
40 C=128+INT(RND(1)*128)
50 D=128+INT(RND(1)*128)
60 E=INT(RND(1)*2)+1
70 F=2*E

```

```

75 POKE36878,F-1

```

```

90 FORH=1T05*E

```

```

100 POKE36874,A

```

```

110 NEXTH

```

```

120 FORH=1T05*E

```

```

130 POKE36875,B

```

```

140 NEXTH

```

```

150 FORH=1T05*E

```

```

160 POKE36876,C

```

```

170 NEXTH

```

```

180 FORH=1T05*E

```

```

190 POKE36877,D

```

```

200 NEXTH

```

```

210 FORH=1T02*E

```

```

220 POKE36874,A

```

```

230 NEXTH

```

```

240 FORH=1T05*E

```

```

250 POKE36876,C

```

```

260 NEXTH

```

```

500 RUN

```

```

1000 PRINT"J":REM CLEAR

```

```

1010 POKE36879,16*(INT(RND(1)*16))+(INT(RND(1)*8))+8

```

```

1020 FORJ=1T05

```

```

1030 PRINT

```

```

1040 NEXT

```

```

1050 PRINT"SYMPHONY FOR A

```

```

1060 PRINT

```

```

1065 FORJ=1TORND(1)*20:PRINT" ";NEXT

```

```

1070 PRINT CHR$(119+INT(RND(1)*15))

```

```

1075 FORJ=1TORND(1)*20:PRINT" ";NEXT

```

```

1080 PRINT CHR$(119+INT(RND(1)*15))

```

```

1085 FORJ=1TORND(1)*20:PRINT" ";NEXT

```

```

1090 PRINT CHR$(119+INT(RND(1)*15))

```

```

1095 FORJ=1TORND(1)*20:PRINT" ";NEXT

```

```

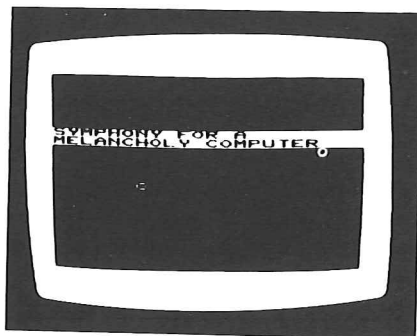
1100 PRINT CHR$(119+INT(RND(1)*15))

```

```

1110 RETURN

```



```

MELANCHOLY COMPUTER "

```

# VIC CHECKERS

The VIC puts up a challenging defence in this game of checkers, or draughts. The game follows all the traditional rules, except that there; is no huffing (penalty for non-capture). Despite this, the VIC will always capture when it can.

The VIC is the solid red circles playing down the board, and you are the open circles playing from the bottom of the screen. You move by entering the letter across the bottom, then the number down the side (as C7), then press RETURN, followed by the letter across the bottom, and the number down the side of the square into which you want to move. If you make a capture, you'll be asked JUMP AGAIN? Enter any letter, then press RETURN if you can jump again, or just press RETURN if you cannot jump again. You'll then be given an additional move. The VIC makes its multiple jumps automatically.

The VIC always has first move. The number of pieces both of you have captured is shown above the board, the game will end if either of you manages to capture all of the other's pieces, or if the VIC judges the situation to be hopeless, and concedes the game. Although the VIC is not always too good at spotting threats to its pieces, it is most reluctant to move into danger.

The screen will clear from time to time during a game, to stop the area under the board being cluttered by inputs. Although there is no mechanism within the program to stop you cheating, there is little to gain by doing so, unless you want to force the computer into a particular position to see how it will react. The VIC is an honourable opponent, and will not cheat.

Kings are made automatically. Your kings are letter K's, and the computer's kings are diamonds. Due to shortage of memory, there is no input validation, so be careful when you enter your moves. A mistake when entering a move can wreck a game. Lines 390 to 405 are the ones in which the VIC checks the safety of its intended moves, predicting captures which you will make in a following move if it makes the contemplated move.

The program has been written in such a way that it is easy to change the characters used on the screen to represent the pieces. All you have to change is lines 750 and 900 to use your own choice of characters for the pieces.

```
10 REM CHECKERS
40 PRINT"PLEASE STAND BY"
50 GOSUB950
60 Z=24
70 Q=0
90 FORG=69TO72: IFA(G)=-1THENA(G)=-2
100 NEXT
110 GOSUB740
130 IFA(Z)=90RA(Z)<1THENGOTO220
140 IFZ<28ANDA(Z)=1THENA(Z)=2
```

```

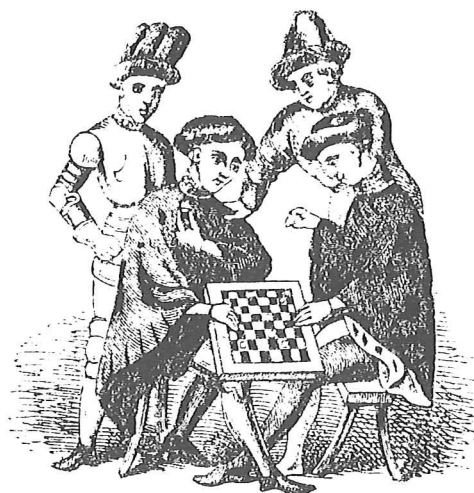
150 Y=1
160 IFA(Z+X(Y))<0AND(A(Z+2*(X(Y)))=0)THENQ=X(Y)
170 IFA(Z)=2AND(A(Z-X(Y))<0AND(A(Z-2*(X(Y)))=0)THENQ=-X(Y)
180 IFQ<0ANDZ+2*Q>23THENGOTO250
190 Q=0
200 IFY=2THEN220
210 Y=2:GOTO160
220 Z=Z+1
230 IFZ<73THEN130
240 IFQ=0THEN340
250 A(Z+Q)=0:A(Z+2*Q)=A(Z):A(Z)=0
260 Z=Z+2*Q:CQ=CQ+1:GOSUB740
270 Q=0:Y=1
280 IFA(Z+X(Y))<0AND(A(Z+2*(X(Y)))=0)THENQ=X(Y)
290 IFQ<0ANDZ+2*Q>23THEN250
310 IFY=1THENY=2:GOTO280
320 GOTO480
340 U=0:Q=0
350 Z=24+INT(RND(1)*49):U=U+1
360 IF(A(Z)=9OR(A(Z)=-1OR(A(Z)=-2OR(A(Z)=0)ANDU<1000)THEN350
380 Y=1
385 IFA(Z+X(Y))<0THEN420
390 IFA(Z+X(Y))=0AND(A(Z+2*(X(Y)))=-1AND(A(Z+2*(X(Y)+1))=-1AND(A(Z+2*(X(Y)-1))=-1)THENQ=X(Y)
395 IFA(Z+X(Y))=0ANDU>150THENQ=X(Y)
400 IFA(Z)=2AND(A(Z-X(Y))=0AND(A(Z-2*(X(Y)))=-1AND(A(Z-2*(X(Y)+1))=-1)THENQ=-X(Y)
405 IFU>600AND(A(Z)=2AND(A(Z-X(Y))=0)THENQ=-X(Y)
410 IFQ<0THEN450
420 IFY=1THENY=2:GOTO390
430 IFU<1000THEN350
440 PRINT"I CONCEDE THE GAME":STOP
450 A(Z+Q)=A(Z):A(Z)=0
470 GOSUB740
480 PRINT"R":INPUT"FROM":A$:INPUT"TO":B$
490 IFU#<>" "THENPRINT"J":U$=""
520 FORW=1TO2:Z=0
530 IFW=1THENC$=A$
540 IFW=2THENC$=B$
550 Z=-24*(C$="G9")-25*(C$="E9")-26*(C$="C9")-27*(C$="A9")-30*(C$="H8")-31*(C$="F8")
552 IFZ<0THEN580
554 Z=-32*(C$="D8")-33*(C$="B8")-37*(C$="G7")-38*(C$="E7")-39*(C$="C7")-40*(C$="A7")
556 IFZ<0THEN580
558 Z=-43*(C$="H6")-44*(C$="F6")-45*(C$="D6")-46*(C$="B6")-50*(C$="G4")-51*(C$="E4")
560 IFZ<0THEN580
562 Z=-52*(C$="C4")-56*(C$="H3")-57*(C$="F3")-58*(C$="D3")-59*(C$="B3")-63*(C$="G2")
564 IFZ<0THEN580
566 Z=-64*(C$="E2")-65*(C$="C2")-66*(C$="A2")-69*(C$="H1")-70*(C$="F1")
568 IFZ<0THEN580
570 Z=-71*(C$="D1")-72*(C$="B1")
580 IFW=1THENZ=Z
590 IFW=2THENZ=Z
600 NEXT:A$="":B$="":C$=""
620 A(E)=A(D):A(D)=0
650 IFABS(D-E)>7THENA((D+E)/2)=0:HU=HU+1
670 GOSUB740
680 IFABS(D-E)>7THENINPUT"JUMP AGAIN":U$:IFU#<>" "THEN480
690 IFHU<12ANDCO<12THEN60
700 IFHU=12THENPRINT"YOU WIN":STOP
710 IFCO=12THENPRINT"I WIN":STOP
740 FORM=24TO72
750 A(M)=-209*(A(M)=1)-218*(A(M)=2)-32*(A(M)=0)-215*(A(M)=-1)-75*(A(M)=-2)-9*(A(M)=9)
760 NEXT

```

```

770 PRINT"♚":PRINTTAB(5)"♚ME"CO"  ♚YOU"HU
780 PRINT:PRINT:PRINT
785 PRINTTAB(6)"♚ABCDEFGH"
790 T=-2:FORK=0T03:PRINTTAB(6)::FORJ=0T03
800 PRINT"♚♚♚":CHR$(A(72-J-13*K)):NEXT:T=T+1
820 PRINT"♚"INT((J+K)/2)+T
830 FORJ=0T03:PRINTTAB(6)"♚":CHR$(A(66-J-13*K)): "♚♚":
850 NEXT:T=T+1
860 PRINT"♚"INT((J+K)/2)+T

```



```

870 NEXT
880 PRINTTAB(6)"♚ABCDEFGH":PRINT
885 REM PRINT"0"0"  Z"Z"  U"U
890 FORM=24T072
900 A(M)=-A(M)=209)-2*(A(M)=218)+0*(A(M)=32)+(A(M)=215)+2*(A(M)=75)-9*(A(M)=9)
910 NEXT
920 RETURN
950 DIMA(99):X(1)=-6:X(2)=-7
960 FORZ=1T099:A(Z)=9:NEXT
970 FORZ=1T032:READB:READC:A(B)=C:NEXT
980 DATA72,1,71,1,70,1,69,1,68,1,67,1,66,1,65,1,64,1,63,1,59,1,58,1,57,1,56,1
990 DATA53,0,52,0,51,0,50,0,46,0,45,0,44,0,43,0
1000 DATA40,-1,39,-1,38,-1,37,-1,33,-1,32,-1,31,-1,30,-1,27,-1,26,-1,25,-1,24,-1
1010 CO=0:HU=0
1015 POKE36879,25
1020 PRINT"♚":RETURN

```

# JELLY BEAN SPACE SWARM

In Adam Burbidge's program, you are the navigator on a starship. Suddenly you find yourself embroiled in an asteroid storm. Only you can Save The Ship From Destruction.

You are the purple checkerboard in the left hand corner of the screen. As you'll see when you run it, the asteroids appear at different points. By using your two controls (4 - left, 6 - right) you have to steer out of the way of the asteroids (which look somewhat like jellybeans, hence the title). The pressure increases as the game continues. The number of asteroids increases, and their colour changes.

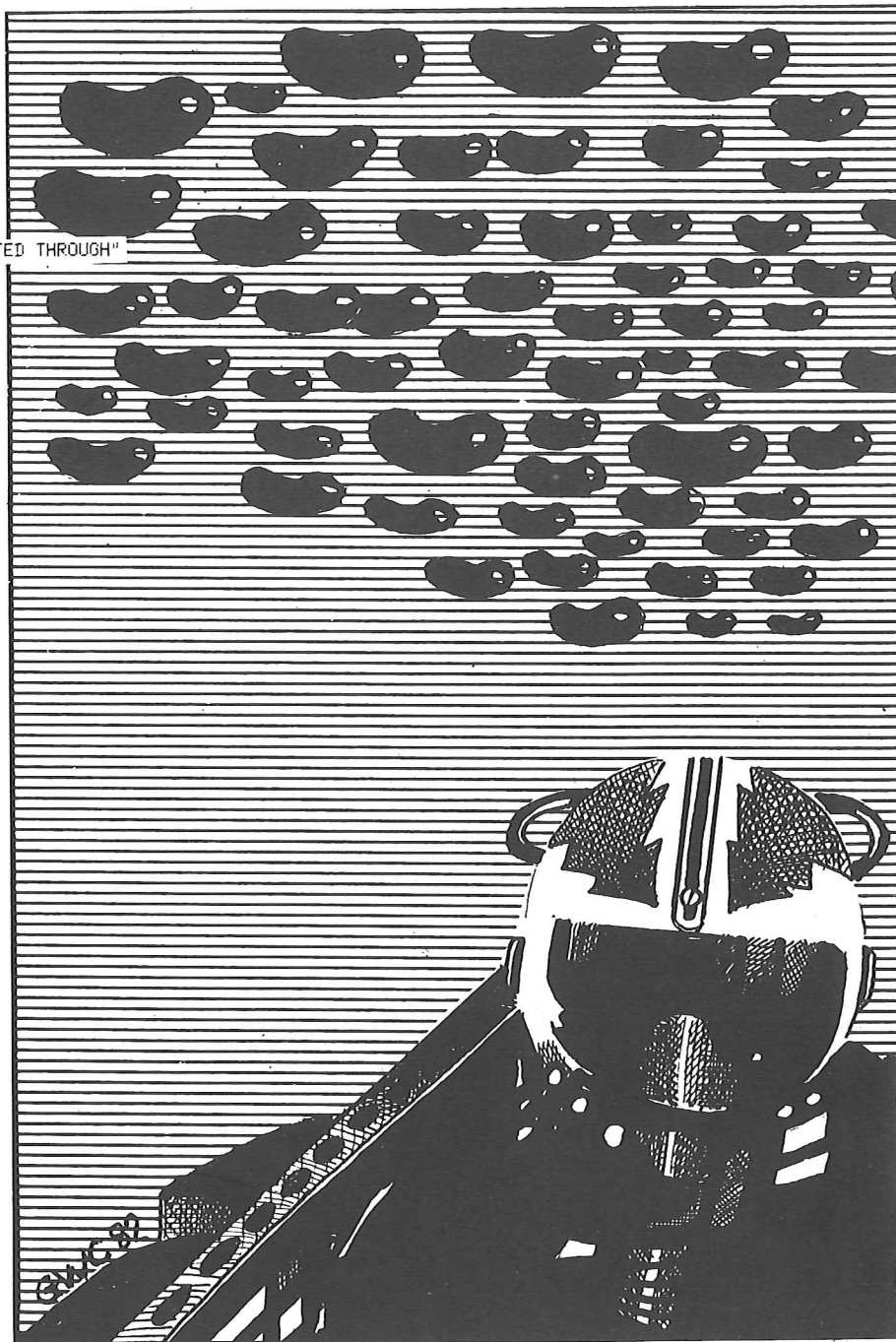
```
1 REM JELLYBEAN SPACE SWARM
2 REM BY ADAM BURBIDGE
3 REM STANDARD MEMORY ONLY
4 SC=0:HS=0
5 PRINT"J":REM CLEAR SCREEN
10 FORJ=1TO20:PRINT:NEXT
11 POKE36879,255
40 POKE7746,102
43 B=38466
45 POKE38466,4
50 FORT=1TO300:NEXT
60 A=7746
65 POKEA,32
70 GET A$:IF A$="4" THEN A=A-1:B=B-1
71 IFA$="6" THEN A=A+1:B=B+1
72 POKE(A-23),32
73 POKE(A-21),32
74 POKE(A-22),32
75 POKEA,102
76 POKEB,4
77 FORT=1TO100:NEXT
78 IFPEEK(A+22)=81 THEN 200
79 SC=SC+1
80 X=INT(RND(1)*506)+7680
89 POKEX+30720,6
90 POKEX,81
93 IF SC>400 THEN 500
95 IF SC>300 THEN GOSUB 400
96 IF SC>200 THEN GOSUB 350
100 PRINT
110 GOTO65
200 POKE 36877,220
210 FORL=15TO0 STEP -1
220 POKE 36878,L
230 FORM=1TO300
240 NEXTM
250 NEXTL
```



```

260 POKE 36877,0
270 POKE 36878,0
280 PRINT "SHIP DESTROYED!!!!"
285 PRINT "SCORE=";SC
286 IF SC>HS THEN PRINT "HIGH SCORE=";SC:HS=SC
287 IF SC<HS THEN PRINT "HIGH SCORE=";HS
290 FOR T=1 TO 7000: NEXT
295 SC=0
300 GOTO 5
350 Y=INT(RND(1)*506)+7680
360 POKEY,81
370 POKEY+30720,2
380 RETURN
400 D=INT(RND(1)*506)+7680
410 POKED,81
420 POKED+30720,5
430 RETURN
500 PRINT "WELL DONE, YOU'VE"
510 PRINT "SUCCESSFULLY NAVIGATED THROUGH"
520 POKE 36878,13
530 POKE 36876,135
540 FOR Q=241 TO 135 STEP -1
550 FOPM=1 TO 100
555 POKE 36876,0
560 NEXT M
570 NEXT Q
580 POKE 36876,0
590 POKE 36878,0
600 END

```



# HANGMAN I

This game needs two players, one to enter a word of her choice and the other player to attempt to guess the word. The VIC draws a dash for each letter, and then changes this into any letter which has been guessed correctly.

## Structure of the program

Line 50: Sets up arrays if needed.

Lines 100 to 130: Accepts the secret word, a letter at a time.

Line 135: Counts for the 10 guesses.

Lines 145 to 170: Select colours at random using the subroutine from 9010, prints the letter if it has been guessed, if not prints a dash.

Lines 210 to 260: Accepts a guess and compares it, letter for letter with the secret word.

Line 270: Sends control to winning message if word is guessed.

Line 300 to 330, then from 1045: Losing message, reveals word.

Lines 1055 to 1100: Offer new game.

Routine from 5000: Adds some sound to brighten things up.

Routine from 9000: Assigns the colour controls to string A\$, and then when subsequently called (as GOSUB 9010) changes the next thing to be printed to a random colour. This subroutine also calls the sound subroutine, then flashes (line 9025) given the border a random colour.

## Suggestions for improvement

- Get the program to draw a hanged man, part by part, as the number of guesses is used up.
- Store a number of words in a DATA statement, so the VIC can choose its own words.
- Note that the random colour choice, the sound routine from 5000 and the random border flash are all effective aids to add to your own programs to brighten them up.

```
10 REM HANGMAN
20 PRINT "J":GOSUB9000
30 INPUT "HOW MANY LETTERS";N
50 IFN>10 THEN DIM B(N),D(N)
100 FORA=1TON:INPUTC$
120 B(A)=ASC(C$):D(A)=B(A)
130 NEXT
135 FORJ=1TO10:PRINT"J"
140 PRINT:PRINT:PRINT
145 FORE=1TON
150 GOSUB9010:PRINTB$;
155 IFB(E)=D(E) THEN PRINT"-";
160 IFB(E)<>D(E)THEN PRINT CHR$(B(E));
170 NEXTE
180 PRINT:PRINT
190 H=0
200 GOSUB9010:PRINTB$; "ENTER GUESS NO."J
210 INPUTC$
220 F=ASC(C$)
230 FORG=1TON
240 IFD(G)=F THEN D(G)=0
250 IFD(G)=0THEN H=H+1
260 NEXTG
270 IFH=NTHEN 1000
280 NEXTJ
290 PRINT
300 GOSUB9010:PRINT"SORRY, TIME IS UP"
310 PRINT
320 GOSUB9010:PRINTB$;"THE WORD WAS ";
330 GOTO1045
```

```

1000 REM WIN
1010 PRINT
1020 GOSUB9010:PRINTB$;"YOU GOT IT IN"J
1040 PRINT
1045 FORQ=1TON:PRINTCHR$(B(Q));:NEXT
1050 PRINT
1055 GOSUB9010:PRINTB$;"ANOTHER GAME";
1060 INPUTC$
1070 IFASC(C$)=ASC("Y")THEN RUN
1080 GOSUB9010:PRINTB$;"OK, THANKS FOR"
1090 GOSUB9010:PRINTB$;"PLAYING!"
1095 FORT=1TO300:NEXT
1097 PRINT
1100 GOTO1080
5000 REM SOUND
5010 POKE36878,INT(RND(1)*8)+8
5020 FORT=1TO100STEP(INT(RND(1)*8)+1)
5030 POKE36876,240-T
5040 POKE36877,T
5060 NEXT
5070 POKE36876,0
5080 POKE36877,0
5090 RETURN
8990 END
9000 A$="HANGMAN"
9010 B$=MID$(A$,INT(RND(1)*7)+1),1)
9020 GOSUB5000
9025 POKE36879,INT(RND(1)*6)+26
9030 RETURN

```

# HANGMAN II

Here's another version of HANGMAN. In this one, the computer picks the word for you to guess. Change the words in the DATA statements from line 370 to add your own vocabulary. The number of 'lives' you have is related to the number of letters in the word to be guessed.

```

10 REM HANGMAN-11
20 FORG=1TORND(1)*22+1
30 READA$
40 NEXT
50 Y=0
60 N=LEN(A$)
80 FORG=1TON
90 B(G)=ASC(MID$(A$,G,1))
100 D(G)=B(G)
110 NEXT:PRINT"███"
120 Q=INT((N+N/2+.5):PRINT"YOU HAVE"Q"CHANCES"
130 FORJ=1TOQ:Y=Y+1
140 GOSUB410
150 IFH=NTHE300
190 PRINT"███"Q+1-J"CHANCES LEFT"
200 INPUTC$
210 F=ASC(C$)

```

```

220 FORG=1TON
230 IFD(G)=FTHENEND(G)=0:J=J-1
240 NEXT: NEXT
250 GOSUB410
260 PRINT:PRINT"SORRY. TIME'S UP!"
280 GOTO330
300 REM**WIN**
310 PRINTTAB(5)"*****LIVES LEFT*****"
320 PRINT"YOU GOT THE WORD IN"Y-1
330 PRINT"IT WAS "A$
340 GOTO340
370 DATA"TERROR","POSTURE","ELEPHANT","STATUS","BACHELOR","ANSWER"
380 DATA"TENOR","BANANA","FIGURE","IDIOT","NARCOTIC","PATHETIC"
390 DATA"WIZARD","WICKED","WIZENED","EVIL"
400 DATA"PARTICLE","ATOM","ELECTRON","START","FAMISHED","EAGLE"
410 H=0
415 PRINT"0000"
417 FORG=1TON
420 IFB(E)=D(E)THENPRINT"0";
430 IFB(E)<>D(E)THENPRINT"X";CHR$(D(E));H=H+1
440 NEXT:PRINT
460 PRINT:IFH<NTHENPRINTH"CORRECT LETTERS"
480 PRINT
490 RETURN

```

# ANTI HANG

This is HANGMAN in reverse. The computer tries to guess the word you have chosen.

When you run the program, you'll be asked HOW LONG IS THE WORD YOU HAVE CHOSEN? Enter the number of letters in the word – if your word was APPLE you'd enter 5 – and then press RETURN. The next thing you'll see on the screen is five inverse blue dashes, one for each letter in the word, as well as LIVES LEFT – 10. This is followed by I GUESS and the letter the computer has guessed. If the letter is one which *is* in the word, say A, then you enter 1, to show it was the first letter in your chosen word. If the letter is not in your word, you enter 0. Assuming the computer guesses A as the first letter. you enter 1, then press RETURN. Immediately, the five inverse blue dashes are reprinted, but this time the first one has changed into an A. The words I GUESS A are still there, to cater for double letters. As there are not two A's in APPLE, you enter 0 and press RETURN.

Here's how one game I played with the VIC progressed. I was thinking of the word START:

T – 5	LIVES LEFT – 10	– – – – T
T – 2	LIVES LEFT – 10	– T – – T
T – 0	LIVES LEFT – 10	– T – – T
E – 0	LIVES LEFT – 9	– T – – T
A – 3	LIVES LEFT – 9	– T A – T
A – 0	LIVES LEFT – 9	– T A – T

O - 0	LIVES LEFT - 8	- T A - T	
R - 4	LIVES LEFT - 8	- T A R T	
R - 0	LIVES LEFT - 8	- T A R T	
S - 1	LIVES LEFT - 8	S T A R T	I WIN I WIN I WIN

You'll find that, generally, the longer the word, the better the chance the VIC has of working it out. ANTIHANG is based on a program written by Toni Baker.

```

10 REMETAORHISHDLFCMUGYPWBJKQXVZ
20 L=10:Q=45
30 PRINT"#####HOW LONG IS THE WORD YOU HAVE CHOSEN?"
40 INPUTN:PRINT"J"
50 DIMA(26),C(N),G(N)
110 FORZ=1TO26
120 A(Z)=PEEK(4101+Z)
130 IFZ<N+1THENG(Z)=Q
140 NEXT
150 Z=INT(RND(0)*3)+1
160 B#=CHR$(A(Z))
170 FORJ=2TO25
180 A(J)=A(J+1)
190 NEXT
200 A=0
210 PRINT"#####132";
220 FORZ=1TON
230 PRINTCHR$(G(Z));
240 NEXT:PRINT
250 PRINT"#####LIVES LEFT -"L
290 PRINT"### GUESS "B#
300 INPUTB1
310 IFB1=0THEN350
320 A1=1
330 G(B1)=ASC(B#)
340 GOTO200
350 F=0
360 FORZ=1TON
370 IFG(Z)=0THENF=1
380 NEXT
390 IFF=0THENPRINT"#####I WIN!!":GOTO390
400 IFA1=0THENL=L-1
410 A1=0:IFL>0THEN150
420 PRINT"#####YOU WIN!!":GOTO420

```

# FLIP

FLIP is an intriguing game, which provides quite a bit of mental stimulation. On a three by three grid are randomly solid (filled in) and empty circles. By flipping them (see the following for an explanation of the flip), you have to end up with a single empty circles. By flipping them (see the following for an explanation of the flip), you have to end up with a single empty circle in the middle position, and solid circles in the other eight positions.

You can only flip a solid circle. You move by entering the number of the piece you wish to flip. Flipping a corner piece causes those adjoining it to change to their opposites (that is, a solid circle opens up, an open circle solidifies). Flipping a

middle circle on one side changes the two either side of it, and flipping the middle one changes the middle piece on all four sides. The piece you flip also changes.

The number of moves you have taken so far is displayed. At the end of the game, the program will pause to tell you how many moves it took to solve it, and then you'll be given a new starting position.

```

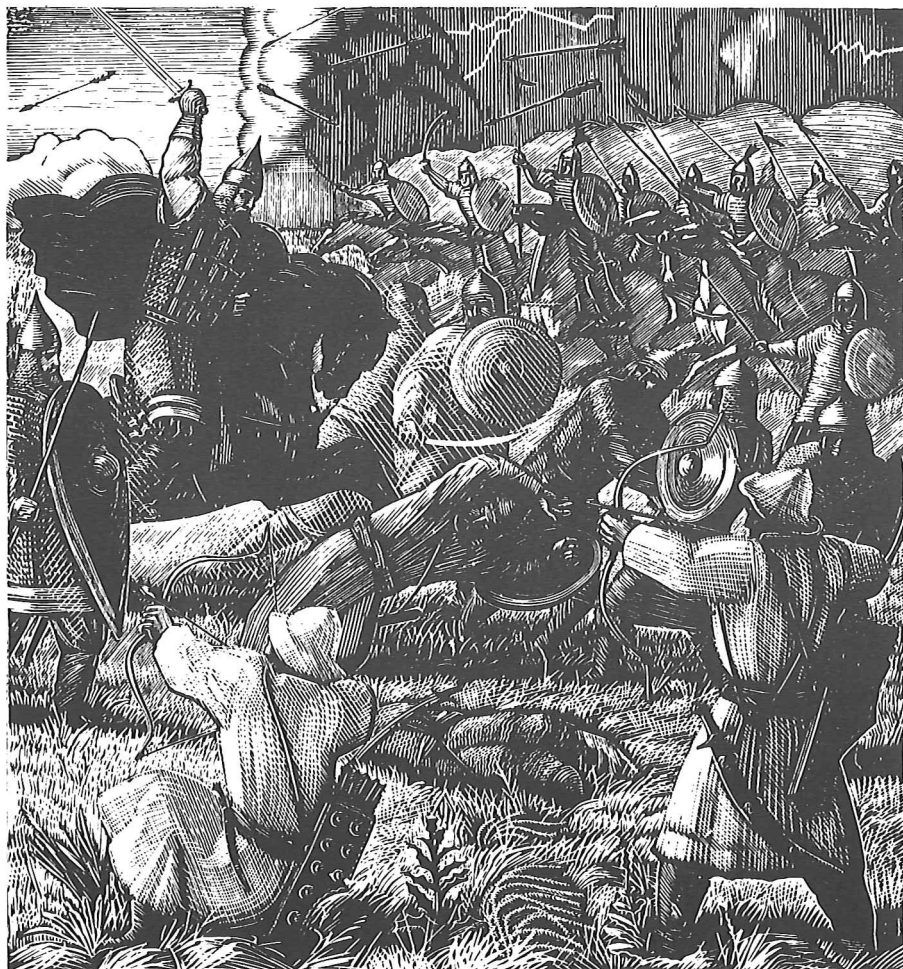
5 REM FLIP
10 M=0:Q=209:X=215
20 FORC=1TO9
30 B=INT(RND(1)+.5)
35 A(C)=Q:IFB=0THENA(C)=X
40 NEXTC
50 GOSUB180
60 N=0
70 FORC=1TO9
80 IFA(C)=XTHENN=N+1
90 NEXTC
100 IFN=1AND A(5)=XTHENGOTO270
110 M=M+1
115 PRINT"MOVE NO."M:PRINT:PRINT
120 PRINT"WHICH ONE TO CHANGE"
125 GETA$:IFA$=""THEN125
126 N=VAL(A$):IFN<1ORND>9THEN125
130 GOSUB310
140 GOTO50
170 END
180 PRINT"C"
190 PRINT:PRINT:PRINT
200 PRINT"1" "2" "3" "CHR$(A(1))" "CHR$(A(2))" "CHR$(A(3))"
210 PRINT
220 PRINT"4" "5" "6" "CHR$(A(4))" "CHR$(A(5))" "CHR$(A(6))"
230 PRINT
240 PRINT"7" "8" "9" "CHR$(A(7))" "CHR$(A(8))" "CHR$(A(9))"
250 PRINT
260 RETURN
270 PRINT"SOLVED IN"M
280 FORT=1TO5000
290 NEXT
300 RUN
310 IFA(N)=XTHENRETURN
320 IFN=1THENF(1)=2:F(2)=4:F(3)=5:F(4)=10
330 IFN=2THENF(1)=1:F(2)=3:F(3)=10:F(4)=10
340 IFN=3THENF(1)=2:F(2)=5:F(3)=6:F(4)=10
350 IFN=4THENF(1)=1:F(2)=7:F(3)=10:F(4)=10
360 IFN=5THENF(1)=2:F(2)=4:F(3)=8:F(4)=6
370 IFN=6THENF(1)=3:F(2)=9:F(3)=10:F(4)=10
380 IFN=7THENF(1)=4:F(2)=5:F(3)=8:F(4)=10
390 IFN=8THENF(1)=7:F(2)=9:F(3)=10:F(4)=10
400 IFN=9THENF(1)=8:F(2)=5:F(3)=6:F(4)=10
410 POKE36878,15
420 FORG=1TO4
425 POKE36879,24+RND(1)*8
427 POKE36875,130+RND(1)*100
430 IFA(F(G))=XTHENA(F(G))=Q:GOTO450
440 IFA(F(G))=QTHENA(F(G))=X
450 NEXTG
455 POKE36875,0:POKE36878,0
460 A(N)=X
470 RETURN

```

# BATTLE

In this board game, which is somewhat like checkers (except for the size of the board, and the method of capture), you are at the bottom of the screen playing up, and the computer is at the top playing down.

The computer always has first move. The aim, as in checkers, is to capture your opponent's men, but unlike checkers you capture by landing on a piece, rather than by jumping over it. You move responding to the prompt FROM? With the number



across the bottom of the square you wish to move from, then a comma, then the number along the side. You do the same for the 'to' square. A move, then, might be:  
FROM? 6, 2

TO? 5, 3

The winner is the first player to capture six of the opponent's pieces.

BATTLE was written by Chris Callender, of Cove, Helensburgh.

```

5 REM BATTLE - CHRIS CALLENDER
10 DATA"123456789"
20 DATA"100000001"
30 DATA"200000002"
40 DATA"300000003"
50 DATA"400000004"
60 DATA"500000005"
70 DATA"600000006"
80 DATA"700000007"
90 DATA"800000008"
100 DATA"123456789"
105 HS=0:CS=0
110 DIMS$(12,13)
120 FORA=1TO10
130 READB$
140 FORB=1TO11:S$(A,B)=MID$(B$,B,1):NEXTB:NEXTA
141 IFINT(RND(1)+.5)=0THEN145
142 S$(5,5)="●":S$(5,7)=" "
145 POKE36879,25:PRINT"J"
150 GOSUB1130
155 PRINT:PRINT:PRINT
157 IFCS=6THENPRINT"I WIN!":STOP
160 INPUT"FROM":A,B
170 INPUT"TO":C,D
180 IFABS(A-C)=1ANDABS(B-D)=1THEN220
190 PRINT"ILLEGAL MOVE":GOTO160
220 IFS$(C+1,D+1)="●"THENHS=HS+1
230 S$(A+1,B+1)=" ":S$(C+1,D+1)="♣"
240 GOSUB1130
245 PRINT:PRINT:PRINT          ":PRINT"
247 IFHS=6THENPRINT"YOU WIN!":STOP
250 A$="♣":GOSUB1000
260 IFFL=1THEN300
270 A$=" ":GOSUB1000
300 S$(E,F)=" "
305 IFS$(E+G,F+H)="♣"THENCS=CS+1
310 S$(E+G,F+H)="●"
320 GOTO150
1000 M E=2:F=2:G=0:H=0
1010 FL=0
1020 IFS$(E,F)<>"●"THEN1100
1040 IFS$(E+1,F+1)=A$THENG=1:H=1
1050 IFS$(E+1,F-1)=A$THENG=1:H=-1
1060 IFS$(E-1,F+1)=A$THENG=-1:H=1
1070 IFS$(E-1,F-1)=A$THENG=-1:H=-1
1080 IFG<0ANDH<0THENFL=1:RETURN
1100 E=E+1:IFE>10THENE=2:F=F+1
1110 IFF>11THENRETURN
1120 GOTO1010
1130 PRINT"♠":FORA=1TO10:PRINT:FORB=1TO11
1140 PRINT"♠":IFS$(A,B)="●"THENPRINT"♠";
1145 IFS$(A,B)="♠"THENPRINT"♠";
1150 IFS$(A,B)="♣"THENPRINT"♣";
1155 PRINTS$(A,B);
1160 NEXT:NEXT
1170 PRINT:PRINT:PRINT"NAME:"CS,"YOU:"HS"
1180 RETURN

```



# CAVEMAN

As the Caveman, you must survive in the labyrinth-like cave system for 25 minutes, battling monsters and finding treasure, while trying to avoid quicksand, and blundering into walls and blocked caves. You have five arrows, and each monster you kill gains you extra wealth. There are also magic caves, which move you randomly through the system.

This program occupies all but 70 bytes of the memory on the standard VIC so there is no room for elaboration. To make sure as many features as possible could be squeezed in, PRINT statements are terse to the point of rudeness.

You start in cave 55. You are shown a view of the system from above at the beginning of the program, and from time to time while a game is underway. The key to the cave's contents is: H – you (for human); \$ – gold; Q – quicksand; a diamond shape – a magic cave; a grey square – a blocked cave; and a full stop – an empty cave.

You can move up, down, right or left one square at a time (which you do by entering N – north, S – south, E – east or W – west). You can also choose to shoot into an adjoining cave, which you do by entering an F. Your cave computer will warn you when things are nerby (NEARBY IS... GOLD is one typical message), but being a Cromagnon VIC 20BC, it tends to malfunction so you only know of the contents of one of the surrounding caves, rather than what is in all of them. As well, it does not tell you which of the surrounding caves it is talking about.

A cave is emptied after you leave it, so you cannot revisit a cave containing gold over and over again to enrich yourself. If a beastie gets you, you fall into quicksand, or run out of air, the game ends with a printout of the system.

When you run the program, you'll see the words CAVE 55 at the top of the screen. If your Cromagnon VIC 20BC has any message for you, it will appear next, followed by a number which is the minutes of air you have left, and then a prompt questionmark. The computer is waiting for your direction input (N, S, E or W) or a decision to fire an arrow (F). If you try to go where there is no path, the message BLOCKED CAVE will appear, and you'll have to enter a new direction. If you say you wish to fire an arrow when your five are used up, you'll be told you're out of arrows, and asked for a new instruction. Good hunting, Caveman.

```
30 DIMA(100):H=0:Q=0:L=0:G=0:AR=5
40 POKE36878,15
50 FORB=1TO100:A(B)=46
60 IFB<12ORB>90OR10*INT(B/10)=BOR10*INT(B/10)=B-1THENA(B)=166
100 NEXT
```

```

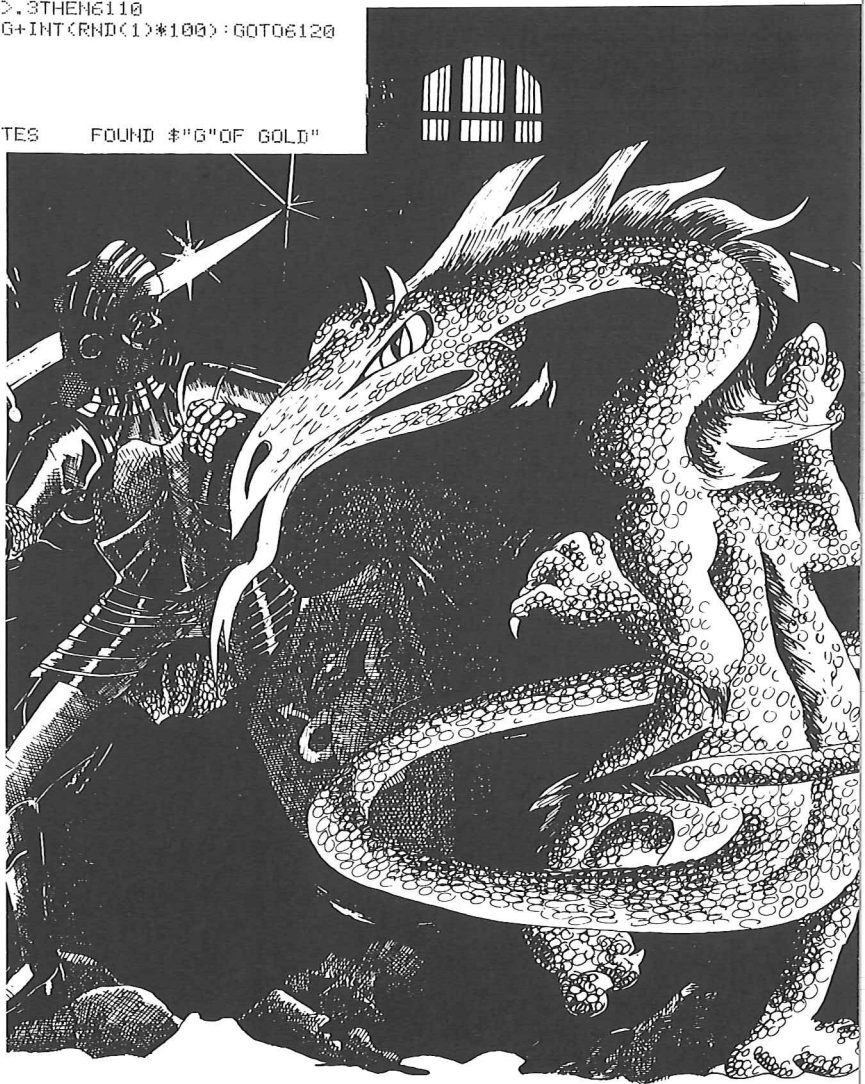
110 FORB=1TO5:RESTORE:FORI=1TO5
115 Z=INT(RND(1)*76)+12:IFA(Z)=166THEN115
120 READC:A(Z)=C
130 NEXT:NEXT
140 DATA166,218,77,81,36
150 FORB=1TO8:READP(B):NEXT:DATA-11,-10,-9,-1,1,9,10,11
230 E=55
240 A(E)=72
245 PRINT"#####":GOSUB9040
250 Q=INT(RND(1)*7)
305 IFQ=0ANDC<>55THENGOSUB9040
310 PRINT"#####CAVE"E
320 IFQ>0THENPRINT"#####G"GOLD"
330 GOSUB5000
340 PRINT"#####25-H
350 INPUTZ$:U=0
360 IFZ$="N"ANDA(E-10)=166ORZ$="S"ANDA(E+10)=166ORZ$="E"ANDA(E+1)=166THENU=1
362 IFZ$="W"ANDA(E-1)=166ORU=1THENPRINT"#####BLOCKED CAVE":FORT=1TO2000:NEXT:GOTO35
0
375 A(E)=46:IFZ$="N"THENE=E-10
380 IFZ$="S"THENE=E+10
390 IFZ$="E"THENE=E+1
400 IFZ$="W"THENE=E-1
405 IFZ$="F"THENGOSUB6000
410 IFA(E)=218THENGOSUB1000
420 IFA(E)=77THENGOSUB2000
430 IFA(E)=81THENGOSUB3000
440 IFA(E)=36THENGOSUB4000
450 H=H+1:IFH=25THENQ=9:GOTO9000
460 FORT=RND(1)*40+130TO177STEPRND(1)*19+2:POKE36875,T:FORM=1TOT:NEXT
465 POKE36876,200-T:NEXT:POKE36875,0:POKE36876,0
470 GOTO250
1000 PRINT"#####MAGIC!!"
1020 FORU=1TO300:POKE36879,RND(1)*24+8:POKE36876,RND(1)*10+240:NEXT:POKE36879,29
1025 POKE36876,0:A(E)=46
1030 E=INT(RND(1)*76+12):IFA(E)=166THEN1030
1040 RETURN
2000 PRINT"#####MONSTER HERE"
2010 FORT=1TO1000:NEXT
2020 M=RND(1):IFM<.2THENPRINT"#####IT RUNS AWAY:RETURN
2030 PRINT"#####IT'S SEEN YOU..."
2040 FORT=1TO1000:NEXT
2050 IFM>.85THENPRINT"AND FLEES"FORT=1TO999:NEXT:RETURN
2060 PRINT"#####AND EATS YOU!!":FORT=1TO990:NEXT:Q=9:GOTO9000
3000 FORJ=1TO20:PRINTTAB(5*J);"#####HORRORS..."
3020 FORU=230TO280STEP10:POKE36879,RND(1)*24+8:POKE36876,100+U/3
3025 NEXT:POKE36879,29:NEXT
3040 PRINT"#####QUICKSAND!!"
3045 POKE36876,0
3050 FORT=1TO1000:NEXT
3055 Q=9
3060 GOTO9000
4000 FORJ=1TO20:PRINTTAB(5*J);"#####WEALTH!!!":FORT=1TO10*J:NEXT:NEXT
4010 K=INT(RND(1)*100)+100
4020 FORU=1TO100:POKE36879,RND(1)*24+8:POKE36876,RND(1)*100+100:NEXT:POKE36879,2
9
4025 POKE36876,0
4040 PRINT"#####GOLD WORTH $"K"!!!":G=G+K:FORT=1TO3000:NEXT:RETURN
5000 Y=1
5005 L=A(E+P(Y))
5006 IFL<>46THEN5020
5007 IFY<8THENY=Y+1:GOTO5006
5010 IFL=46THENRETURN
5020 PRINT"#####NEARBY IS..."

```

```

5030 IFL=166THENPRINT"NO PATH"
5040 IFL=218THENPRINT"MAGIC"
5050 IFL=77THENPRINT"MONSTER"
5060 IFL=81THENPRINT"QUICKSAND"
5070 IFL=36THENPRINT"GOLD"
5080 FORT=1TO3000:NEXT:RETURN
6000 AR=AR-1:IFAR=0THENPRINT"NO ARROWS":RETURN
6010 PRINTAR"ARROWS":SS=0
6020 INPUT"XATO";S$
6030 IFS$="N"AND(A(E-10)=77THENS=1:YT=E-10
6040 IFS$="S"AND(A(E+10)=77THENS=1:YT=E+10
6050 IFS$="E"AND(A(E+1)=77THENS=1:YT=E+1
6060 IFS$="W"AND(A(E-1)=77THENS=1:YT=E-1
6070 IFSS=0THENPRINT"NO LUCK":GOTO6120
6080 PRINT"XHIT"
6090 FORT=1TO999:NEXT:IFRND(1)>.3THEN6110
6100 PRINT"KILLED":A(YT)=46:G=G+INT(RND(1)*100):GOTO6120
6110 PRINT"WOUNDED"
6120 FORT=1TO3000:NEXT:RETURN
9000 IFQ=9THENGOTO9015
9010 PRINT"NO AIR LEFT"
9015 PRINT"XXXXX"SURVIVED"H"MINUTES      FOUND $"G"OF GOLD"
9040 A(E)=72
9050 FORJ=1TO100
9060 PRINTCHR$(A(J)):
9070 IF10*INT(J/10)=JTHENPRINT
9080 NEXT:IFQ=9THENEND
9090 RETURN

```



# Appendices

## Error Messages

**BAD DATA**...String data was received, but program expecting numeric data.

**BAD SUBSCRIPT**...Element of an array outside of the range specified in the DIM statement.

**CAN'T CONTINUE**...The CONT command will not work, either because the program was never RUN, there has been an error, or a line has been edited.

**DEVICE NOT PRESENT**...The required I/O device was not available for an OPEN, CLOSE, CMD, PRINT #, INPUT #, or GET #.

**DIVISION BY ZERO**...Division by zero impossible.

**EXTRA IGNORED**...Too many items of data entered, in response to an INPUT statement. Only the first few items accepted.

**FORMULA TOO COMPLEX**...String expression should be split into at least two parts.

**ILLEGAL DIRECT**...INPUT can only be used within a program, and not in direct mode.

**ILLEGAL QUANTITY**...Number used as argument of function or statement out of allowable range.

**LOAD**...Problem with program on tape.

**NEXT WITHOUT FOR**...Either incorrectly nesting loops, or having a variable name in a NEXT statement that doesn't correspond with one in a FOR statement.

**OUT OF DATA**...A READ statement executed but no data left unREAD in DATA statement.

**OUT OF MEMORY**...No more RAM available. May also occur when too many FOR loops nested, or too many GOSUBs in effect.

**OVERFLOW**...The result of a computation is larger than  $1.70141884E + 38$ .

**REDIM'D ARRAY**...An array may only be DIMensioned once.

**REDO FROM START**...Character data was typed in during an INPUT statement when numeric data was expected. Just re-type the entry so that it is correct, and the program will continue by itself.

**RETURN WITHOUT GOSUB**...RETURN statement encountered, when no GOSUB command issued.

**STRING TOO LONG**...A string can contain up to 255 characters.

**SYNTAX...** A statement is unrecognizable by the VIC. A missing or extra parenthesis, misspelled keywords, etc.

**TYPE MISMATCH...** This error occurs when a number is used in place of a string, or vice-versa.

**UNDEF'D FUNCTION...** A user defined function was referenced, but it has never been defined using the DEF FN statement.

**UNDEF'D STATEMENT...** An attempt was made to GOTO or GOSUB or RUN a line number that doesn't exist.

BORDER								
SCREEN	BLK	WHT	RED	CYAN	PUR	GRN	BLU	YEL
BLACK	8	9	10	11	12	13	14	15
WHITE	24	25	26	27	28	29	30	31
RED	40	41	42	43	44	45	46	47
CYAN	56	57	58	59	60	61	62	63
PURPLE	72	73	74	75	76	77	78	79
GREEN	88	89	90	91	92	93	94	95
BLUE	104	105	106	107	108	109	110	111
YELLOW	120	121	122	123	124	125	126	127
ORANGE	136	137	138	139	140	141	142	143
LT. ORANGE	152	153	154	155	156	157	158	159
PINK	168	169	170	171	172	173	174	175
LT. CYAN	184	185	186	187	188	189	190	191
LT. PURPLE	200	201	202	203	204	205	206	207
LT. GREEN	216	217	218	219	220	221	222	223
LT. BLUE	232	233	234	235	236	237	238	239
LT. YELLOW	248	249	250	251	252	253	254	255

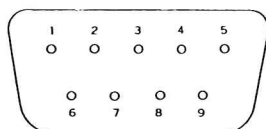
POKE 36879,

SPEAKER COMMANDS:	WHERE X CAN BE:	FUNCTION:
POKE 36878, X	0 to 15	sets volume
POKE 36874, X	128 to 255	plays tone
POKE 36875, X	128 to 255	plays tone
POKE 36876, X	128 to 255	plays tone
POKE 36877, X	128 to 255	plays "noise"

# COLOR CODES MEMORY MAP

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
38400																						
38422																						
38444																						
38466																						
38488																						
38510																						
38532																						
38554																						
38576																						
38598																						
38620																						
38642																						
38664																						
38686																						
38708																						
38730																						
38752																						
38774																						
38796																						
38818																						
38840																						
38862																						
38884																						

## GAME I/O



PIN #	TYPE
1	JOY0
2	JOY1
3	JOY2
4	JOY3
5	POT Y
6	LIGHT PEN
7	+5V
8	GND
9	POT X

# SCREEN CHARACTER CODES

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
7680																						
7702																						
7724																						
7746																						
7768																						
7790																						
7812																						
7834																						
7856																						
7878																						
7900																						
7922																						
7944																						
7966																						
7988																						
8010																						
8032																						
8054																						
8076																						
8098																						
8120																						
8142																						
8164																						









# **Fifty outstanding programs for your VIC-20!**

Whatever sort of fun programs you want for your VIC-20, you'll find them in this great book.

ARCADE GAMES—like *Zombie Island* and *Space Fighter*

ADVENTURE GAMES—like *Caveman* and *Space Trek*

BOARD GAMES—like *Checkers*, *Gomoku* and *Othello/Reversi*

BRAIN STRETCHERS—like *Colourmind*, *Flip* and *Mento*

GAMBLING GAMES—like *Roulette* and *Craps*

**AND MUCH MORE!!**

Write poetry with your VIC, work out your biorhythms, create messages in double-sized characters, drive your racing car on the *Vic Speedway*. . . and more. Don't plan on going out too much in the next few months. You won't be able to tear yourself away from your VIC-20

All programs fit the standard memory VIC-20. Several are supplied with special versions so you can use joysticks if you have them. Non-joystick versions are also included. All games have been extensively 'play-tested'. . . and have passed with flying colours.

**Another great book from INTERFACE**